



King's Research Portal

DOI:

[10.1080/09540091.2016.1243655](https://doi.org/10.1080/09540091.2016.1243655)

Document Version

Peer reviewed version

[Link to publication record in King's Research Portal](#)

Citation for published version (APA):

Spratling, M. W. (2016). A neural implementation of Bayesian inference based on predictive coding. *CONNECTION SCIENCE*, 28(4), 346-383. <https://doi.org/10.1080/09540091.2016.1243655>

Citing this paper

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

General rights

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Research Portal

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

A neural implementation of Bayesian inference based on predictive coding

M. W. Spratling

King's College London, Department of Informatics, London. UK. michael.spratling@kcl.ac.uk

Abstract

Predictive coding is a leading theory of cortical function that has previously been shown to explain a great deal of neurophysiological and psychophysical data. Here it is shown that predictive coding can perform almost exact Bayesian inference when applied to computing with population codes. It is demonstrated that the proposed algorithm, based on predictive coding, can: decode probability distributions encoded as noisy population codes; combine priors with likelihoods to calculate posteriors; perform cue integration and cue segregation; perform function approximation; be extended to perform hierarchical inference; simultaneously represent and reason about multiple stimuli; and perform inference with multi-modal and non-Gaussian probability distributions. Predictive coding thus provides a neural network based method for performing probabilistic computation and provides a simple, yet comprehensive, theory of how the cerebral cortex performs Bayesian inference.

Keywords: Bayes; priors; inference; multisensory integration; function approximation; predictive coding; population coding; neural networks

1 Introduction

It is widely believed that the brain performs Bayesian inference (Chater et al., 2006; Griffiths et al., 2008; Griffiths and Tenenbaum, 2006; Kersten et al., 2004; Knill and Richards, 1996; Lee and Mumford, 2003; Rao et al., 2002; Vilares and Kording, 2011; Yuille and Kersten, 2006). This requires the representation and manipulation of probability distributions. A leading theory of how the brain represents probability distributions suggests that it does so using population coding (Anderson and Van Essen, 1994; Barber et al., 2003; Beck et al., 2011; Deneve et al., 1999, 2001; Földiák, 1993; Ganguli and Simoncelli, 2014; Jazayeri and Movshon, 2006; Latham et al., 2003; Ma et al., 2006, 2008; Pouget et al., 2013, 2003, 2000, 1998; Sanger, 1996; Seilheimer et al., 2014; Zemel et al., 1998). In such a code, the activity across a population of neurons represents a probability distribution: each neuron represents a value of the random variable, and its firing rate encodes the probability associated with that value (although the firing rate may be corrupted with noise). Together, the population of neurons provide a discretely sampled approximation to a continuous probability density function.

There have been a number of previous, neurally-based, accounts of how the brain manipulates such population codes to perform probabilistic inference. For example, models of how probability distributions can be encoded by neural firing rates (Anderson and Van Essen, 1994; Barber et al., 2003; Zemel et al., 1998); models of how priors can be encoded into the receptive fields (RFs) of neurons (Ganguli and Simoncelli, 2010, 2014; Girshick et al., 2011; Shi and Griffiths, 2009); and models of how separate sources of sensory evidence can be combined in a statistically optimal manner. Notable models of the latter type employ a network of integrate-and-fire neurons (Ma et al., 2006, 2008; Seilheimer et al., 2014), or a basis function neural network with attractor dynamics (Deneve et al., 1999, 2001; Latham et al., 2003; Pouget et al., 2003, 2000, 1998). The latter type of network can also be used to perform function approximation with probabilistically defined variable values (Deneve et al., 2001; Pouget et al., 2002). For details of the wide range of different models of probabilistic computation the reader is referred to recent reviews on this topic (Ma, 2012; Ma and Jazayeri, 2014; Pouget et al., 2013; Vilares and Kording, 2011).

In this article an alternative neural network model is proposed for computing with population codes that approximate probability distributions. The motivation for developing this new algorithm was to create a single, biologically-plausible, method capable of performing all the probabilistic inference tasks mentioned in the previous paragraph, and hence, to provide a more comprehensive neural model of probabilistic computation (as performed by the cerebral cortex) than has previously been proposed. The proposed model succeeds in being able to: decode probability distributions encoded as noisy population codes; combine priors with likelihoods to calculate posteriors; perform cue integration; and perform function approximation; Furthermore, it goes beyond the existing algorithms in being able to additionally: simultaneously represent and reason about multiple stimuli; perform inference with non-Gaussian probability distributions; perform cue integration with a non-flat prior; perform cue segregation as well as integration; and perform hierarchical inference.

The proposed algorithm, PC/BC-DIM, is a version of Predictive Coding (PC; Huang and Rao, 2011; Rao and Ballard, 1999) reformulated to make it compatible with Biased Competition (BC) theories of cortical function (Spratling, 2008a,b) and that is implemented using Divisive Input Modulation (DIM; Spratling et al., 2009) as

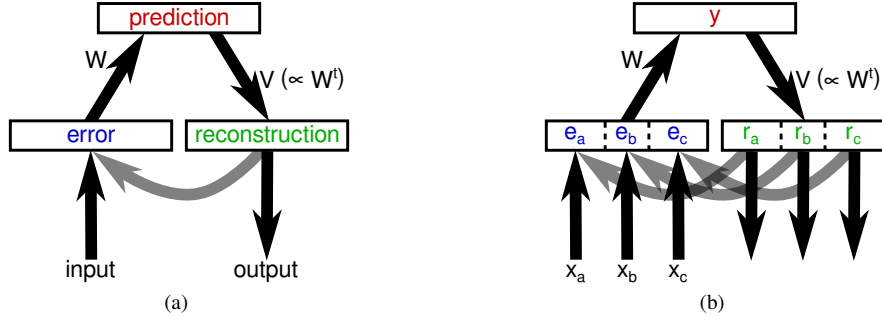


Figure 1: (a) A single processing stage in the PC/BC-DIM neural network architecture. Rectangles represent populations of neurons and arrows represent connections between those populations. The population of prediction neurons constitute a model of the input environment. Individual neurons represent distinct causes that can underlie the input. The belief that each cause explains the current input is encoded in the activation level, y , and is used to reconstruct the expected input given the predicted causes. This reconstruction, r , is calculated using a linear generative model (see equation 1). Each element of the reconstruction is compared to the corresponding element of the actual input, x , in order to calculate the residual error, e , between the predicted input and the actual input (see equation 2). The errors are subsequently used to update the predictions (via the feedforward weights W , see equation 3) in order to make them better able to account for the input, and hence, reduce the error at the next iteration. The weights V are the transpose of the weights W , but are normalised so that the maximum value of each column is unity. The inputs to a processing stage may come from the prediction neurons of this or another processing stage, or the reconstruction neurons of another processing stage, or may be external, sensory-driven, signals. The inputs can also be a combination of any of the above. (b) When inputs come from multiple sources, it is sometimes convenient to consider the population of error neurons to be partitioned into sub-populations which receive these separate sources of input. As there is a one-to-one correspondence between error neurons and reconstruction neurons, this means that the reconstruction neuron population can be partitioned similarly.

the method for updating error and prediction neuron activations. DIM calculates reconstruction errors using division, which is in contrast to other implementations of PC that calculate reconstruction errors using subtraction (Spratling, 2016). The PC/BC-DIM algorithm has previously been shown to explain a large range of neurophysiological and psychophysical data including: orientation tuning, surround suppression and cross-orientation suppression in primary visual cortex (V1; Spratling, 2010, 2011, 2012a), the learning of Gabor-like RFs in V1 (Spratling, 2012c), gain modulation as is observed, for example, when a retinal RF is modulated by eye position (De Meyer and Spratling, 2011, 2013), contour integration (Spratling, 2013b, 2014), the modulation of neural response due to attention (Spratling, 2008a, 2014), and the saliency of visual stimuli (Spratling, 2012b). A second motivation for the current work was to extend the range of phenomena that can be simulated by PC/BC-DIM to include Bayesian inference. The current work also suggests that all the diverse biophysical behaviours that can be explained by PC/BC-DIM may have a single, probabilistic, interpretation.

2 Methods

2.1 The PC/BC-DIM Algorithm

PC/BC-DIM is a hierarchical neural network. Each level, or processing stage, in the hierarchy is implemented using the neural circuitry illustrated in Fig. 1a. A single PC/BC-DIM processing stage consists of three separate neural populations,^a and the behaviour of the neurons in these three populations is determined by the following equations:

$$r = Vy \quad (1)$$

$$e = x \oslash (\epsilon_2 + r) \quad (2)$$

$$y \leftarrow (\epsilon_1 + y) \otimes We \quad (3)$$

^aPrevious work with this algorithm, and with other implementations of predictive coding, have proposed that each processing stage consists of two neural populations: error neurons and prediction neurons. The operation performed by the reconstruction neurons in the current version of PC/BC-DIM was performed within the error neurons in previous versions.

Where \mathbf{x} is a (m by 1) vector of input activations, \mathbf{e} is a (m by 1) vector of error neuron activations; \mathbf{r} is a (m by 1) vector of reconstruction neuron activations; \mathbf{y} is a (n by 1) vector of prediction neuron activations; \mathbf{W} is a (n by m) matrix of feedforward synaptic weight values; \mathbf{V} is a (m by n) matrix of feedback synaptic weight values; ϵ_1 and ϵ_2 are parameters; and \oslash and \otimes indicate element-wise division and multiplication respectively. For all the experiments described in this paper ϵ_1 and ϵ_2 were given the values 1×10^{-6} and 1×10^{-4} respectively. Parameter ϵ_1 prevents prediction neurons becoming permanently non-responsive. It also sets each prediction neuron's baseline activity rate and controls the rate at which its activity increases when an input stimulus is presented within its RF. Parameter ϵ_2 prevents division-by-zero errors and determines the minimum strength that an input is required to have in order to effect prediction neuron response. As in all previous work with PC/BC-DIM, these parameters have been given small values compared to typical values of \mathbf{y} and \mathbf{x} , and hence, have negligible effects on the steady-state activity of the network. The matrix \mathbf{V} is equal to the transpose of the \mathbf{W} , but each column is normalised to have a maximum value of one. Hence, the feedforward and feedback weights are simply rescaled versions of each other. Given that the \mathbf{V} weights are fixed to the \mathbf{W} weights there is only one set of free parameters, \mathbf{W} , and references to the "synaptic weights" refer to the elements of \mathbf{W} . Here, as in previous work with PC/BC-DIM only non-negative weights, inputs, and activations are used.

Initially the values of \mathbf{y} were all set to zero, although random initialisation of the prediction node activations can also be used with little influence on the results. Equations 1, 2 and 3 were then iteratively updated with the new values of \mathbf{y} calculated by equation 3 substituted into equation 1 and 3 to recursively calculate the neural activations. To perform simulations with a hierarchical model, equations 1, 2 and 3 were evaluated for each processing stage in turn (starting from the lowest stage in the hierarchy), and this process was repeated to iteratively update the neural activations in each processing stage at each time-step. If the input remains constant, the network activity will converge to a steady-state. The time taken to reach a steady-state is strongly influenced by the number of synaptic weights. For small networks, like those used in sections 3.1–3.4, five iterations are sufficient. For medium-sized networks, like those used in section 3.5, approximately 20 iterations are sufficient. For large networks, like that used in section 3.6, approximately 50 iterations are required. To ensure each network reached a steady-state, for simulations on small and medium sized networks, the iterative process was terminated after 25 iterations, and for simulations on large networks (and those for the hierarchical network used in section 3.5) 50 iterations were performed. It is the response of the network at the time when the iterative process was terminated that are reported in the results.

PC/BC-DIM is an abstract, functional, model that aims to explore the computational, rather than the physiological, mechanisms which underlie cortical function (Spatling, 2011). However, it is possible to speculate about the potential biological implementation of the model. There are many different ways in which the simple circuitry of PC/BC-DIM model could potentially be implemented in the much more complex circuitry of the cortex (Kok and de Lange, 2015; Spatling, 2008b, 2011, 2012b, 2013a). However, the most straightforward explanation would equate prediction neurons with the sub-population of cortical pyramidal cells (mostly found in cortical layers II and III) whose axon projections form the feedforward connections between cortical regions, and to equate reconstruction neurons with the sub-population of cortical pyramidal cells (mostly found in cortical layer VI) whose axon projections form the feedback connections between cortical regions (Barbas and Rempel-Clower, 1997; Barone et al., 2000; Budd, 1998; Crick and Koch, 1998; Felleman and Van Essen, 1991; Johnson and Burkhalter, 1997; Markov et al., 2014; Mountcastle, 1998). This is consistent with previous work showing that the behaviour of the prediction neurons in the PC/BC-DIM model can explain the response properties of cortical pyramidal cells in both the ventral (Spatling, 2008a, 2010, 2011, 2012a,c, 2014), and dorsal (De Meyer and Spatling, 2011, 2013) pathways of the cortical visual system, and is also consistent with brain imaging data (Alink et al., 2010; Egner et al., 2010; Kok and de Lange, 2015; Kok et al., 2012; Smith and Muckli, 2010; Summerfield and Egner, 2009; Summerfield et al., 2006; Wacongne et al., 2011). It is possible to equate the error-detecting neurons with the spiny-stellate cells in cortical layer IV, which are the major targets of cortical feedforward connections and sensory inputs. However, it is also possible that the error-detection is performed in the dendrites of the superficial layer pyramidal cells (Spatling and Johnson, 2003) rather than in a separate neural population; or via synaptic depression which can produce the specific form of divisive inhibition required by the error-neurons in the PC/BC-DIM model (Rothman et al., 2009); or that the error neurons reside in the thalamus, individual regions of which receive connections from layer VI pyramidal cells (putative reconstruction neurons) as well as either sensory input or input from lower cortical regions (Olsen et al., 2012; Sherman, 2016; Shipp, 2004).

The mechanisms employed by the prediction and error neurons differ from those typically used in artificial neural networks; *i.e.*, linear summation of inputs followed by a nonlinear activation function (Rosenblatt, 1962; Rumelhart et al., 1986). Specifically, the prediction neurons perform a multiplication operation and the error neurons perform a division operation. However, both these nonlinear mechanisms are biologically-plausible. Neurons that perform multiplicative (Salinas and Sejnowski, 2001; Salinas and Thier, 2000), and divisive (Carandini and

Heeger, 1994; Heeger, 1992), operations are common throughout the brain. A range of biophysical mechanisms have been proposed to underlie both these nonlinear operations, including the interplay between linear neurons in a network (Brozović et al., 2008; Chance and Abbott, 2000; Murphy and Miller, 2003; Reynolds and Chelazzi, 2004; Salinas and Abbott, 1996), nonlinear dendritic integration (Gabbiani et al., 2002; Jaffe and Carnevale, 1999; Koch and Segev, 2000; Larkum et al., 2004; London and Häusser, 2005; Mehaffey et al., 2005; Mel, 1994; Mitchell and Silver, 2003; Phillips, 2016; Spruston, 2008; Spruston and Kath, 2004; Stuart and Häusser, 2001), and synaptic mechanisms (Alger, 2002; Branco and Staras, 2009; Rothman et al., 2009; Sherman and Guillery, 1998).

In previous work with this algorithm, the reconstruction has been used purely as a means to calculate the errors, and hence, equations 1 and 2 have been combined into a single equation. Thus, the underlying mathematical model is identical to that used in previous work, but the interpretation has changed in order to consider the reconstruction to be represented by a separate neural population. Furthermore, in the current work the reconstruction neurons constitute the output of the model, and provide inputs to other processing stages in a hierarchical model. In contrast, previous work with PC/BC-DIM has used the prediction neurons as the outputs of each processing stage (Spratling, 2008a, 2012c). This is also in contrast to other versions of predictive coding (Friston, 2005; Rao and Ballard, 1999) that have used two sources of output from each processing stage: error neurons for the feedforward connections from lower to higher processing stages, and prediction neurons for the feedback connections.

2.2 Representing Causes and Performing Explaining Away

The values of \mathbf{y} represent predictions of the causes underlying the inputs to the network (*i.e.*, latent variables). The values of \mathbf{r} represent the expected inputs given the predicted causes. The values of \mathbf{e} represent the residual error between the reconstruction, \mathbf{r} , and the actual input, \mathbf{x} . The full range of possible causes that the network can represent are defined by the weights, \mathbf{W} . Each row of \mathbf{W} (which correspond to the weights targeting an individual prediction neuron, or its RF) can be thought of as a “basis vector” or “elementary component” or “preferred stimulus”, and \mathbf{W} as a whole can be thought of as a “dictionary” or “codebook” of possible representations, or as a model of the external environment, or as the parameters of a generative model. The activation dynamics, described by equations 1, 2 and 3, perform gradient descent on the reconstruction error in order to find prediction neuron activations that accurately reconstruct the input (Achler, 2014; Spratling, 2012c; Spratling et al., 2009). Specifically, the equations operate to minimise the Kullback-Leibler (KL) divergence between the input (\mathbf{x}) and the reconstruction of the input (\mathbf{r}) (Solbakken and Junge, 2011; Spratling et al., 2009). Gradient descent can also be implemented using subtraction rather than division to calculate the reconstruction errors (Achler, 2014; Harpur, 1997) as is the case in the Rao and Ballard (1999) version of predictive coding. In this case, gradient descent attempts to find the prediction neuron activations that minimise the sum squared residual error (Achler, 2014; Harpur, 1997). However, the subtractive method typically converges to a solution more slowly, and the solution is less sparse. Furthermore, the subtractive version is less biologically-plausible as it requires error neurons to be able to have negative firing rates,^b and it also successfully simulates far less neurophysiological data (Spratling, 2008a, 2013a).

At the steady-state, the PC/BC-DIM algorithm will have selected a subset of active prediction neurons whose RFs (which correspond to basis functions) best explain the underlying causes of the sensory input. The strength of activation, \mathbf{y} , reflects the strength with which each basis function is required to be present in order to accurately reconstruct the input. This strength of response also reflects the probability with which that basis function (the preferred stimulus of the active prediction neuron) is believed to be present, taking into account the evidence provided by the input signal and the full range of alternative explanations encoded in the RFs of the whole population of prediction neurons.

If prediction neurons represent distinct causes such as the presence of different objects in a visual scene (Lochmann and Deneve, 2011), or different odours in an olfactory scene (Beck et al., 2012), then each prediction neuron’s activation represents the probability that its preferred stimulus is present in the input (Spratling, 2008b, 2012c, 2013b, 2014; Spratling et al., 2009). This is consistent with the idea (Achler and Amir, 2008; Anastasio et al., 2000; Barlow, 1969; Deneve, 2008; Lee and Mumford, 2003; Lochmann et al., 2012) that the brain computes with “explicit probability codes” (Ma et al., 2008). The activation dynamics of the PC/BC-DIM algorithm enable the prediction neurons to perform a form of perceptual inference, in which evidence that supports one cause is explained away preventing responses from prediction neurons representing other, less likely, causes (Kersten et al., 2004; Lochmann and Deneve, 2011; Spratling, 2014). In common with other neural networks that can perform explaining away (Beck et al., 2012; Lochmann and Deneve, 2011; Lochmann et al., 2012), PC/BC-DIM employs divisive normalisation that targets the inputs to the network (see equation 2). The mechanism of divisive input

^bIt is possible to re-implement the Rao and Ballard (1999) algorithm using only non-negative firing rates (Ballard and Jehee, 2012), however, this results in a model that is extremely complex and requires a degree of coordination between the actions of different connections that is unlikely to be feasible in a biological system.

normalisation employed by PC/BC-DIM is, however, slightly different as prediction neurons can inhibit their own inputs (in contrast to [Lochmann and Deneve, 2011](#); [Lochmann et al., 2012](#)) and each prediction neuron contributes independently to the strength of inhibition (in contrast to [Beck et al., 2012](#)).

Rather than representing distinct causes, the prediction neurons can alternatively be used to represent possible values of a continuous variable, such as the orientation of a visual stimulus ([Spratling, 2010, 2011, 2012a,b,c, 2013b](#)). In this case, different prediction neurons can be tuned to different values. Each prediction neuron then signals the belief that the input stimulus takes a particular value (*e.g.*, is at a particular orientation), while the population of prediction neurons represent the probabilities for the range of possible values.

Previous work with PC/BC-DIM has explored the ability of the prediction neurons to identify latent causes and has shown that the behaviour of the prediction neurons is consistent with the response properties of cortical pyramidal cells ([De Meyer and Spratling, 2011](#); [Spratling, 2010, 2011, 2012a,c, 2014](#); [Spratling et al., 2009](#)). In contrast, this article explores how the reconstruction neurons can be used to calculate probability distributions, and hence, how PC/BC-DIM can be used to perform Bayesian inference. This interpretation of the PC/BC-DIM algorithm will be described in the next section.

2.3 Computing with Probability Distributions

As described in the preceding section, the weights in a PC/BC-DIM network are basis functions or elementary components that can be combined together to reconstruct the input stimulus. If the inputs to a PC/BC-DIM network are probability distributions, then the weights need to represent the elementary components of such probability distributions so that any specific probability distribution that is presented to the network can be reconstructed from those elementary components. Hence, when applied to population codes that encode probability distributions, the PC/BC-DIM model can be seen to have strong similarities to the kernel density estimate (KDE) model of encoding and decoding probability distributions ([Anderson and Van Essen, 1994](#); [Barber et al., 2003](#); [Zemel et al., 1998](#)). Specifically, in this earlier model it is proposed that a probability distribution can be reconstructed by summing basis functions in proportion to the firing rates of the neurons associated with each basis function. This is the operation performed by equation 1 when the columns of \mathbf{V} are interpreted as basis functions appropriate for representing probability distributions. One difference is that in the current model each basis function is equal (up to a scaling factor) to the synaptic weights of that neuron, whereas in the KDE model the basis functions are not the weights ([Barber et al., 2003](#); [Zemel et al., 1998](#)). It is necessary to find neural firing rates (the y values in the PC/BC-DIM model) appropriate for representing an input probability distribution in terms of basis functions. One method for doing this for the KDE model uses the expectation-maximisation algorithm to find neural responses that minimise the Kullback-Leibler (KL) divergence between the input distribution and the reconstructed distribution ([Zemel et al., 1997, 1998](#)). For the PC/BC-DIM algorithm, prediction neuron firing rates appropriate for reconstructing a probability distribution are found using equations 1 to 3. The PC/BC-DIM algorithm is closely related to the particular method of performing non-negative matrix factorisation (NMF) proposed by [Lee and Seung \(2001\)](#) ([Solbakken and Junge, 2011](#); [Spratling et al., 2009](#)). This form of NMF also minimises the KL divergence. It would also be possible to apply the [Rao and Ballard \(1999\)](#) version of predictive coding, using subtraction to calculate the reconstruction errors, to find the prediction neuron activities. This would find neural responses that minimise the least squares error between the input probability distribution and the reconstructed distribution ([Achler, 2014](#); [Harpur, 1997](#)). This succeeds in reproducing most of the results presented in this article, but not all. There are also other reasons for preferring the PC/BC-DIM implementation of predictive coding, as listed in the preceding section.

If a PC/BC-DIM network reconstructs the probability distribution that is presented to its inputs, then one useful operation that may be performed is to reconstruct a less noisy version of a corrupted input distribution. Experiments demonstrating this ability of the PC/BC-DIM network are described in section 3.1. However, simply reconstructing the input distribution is otherwise not very useful. For example, it does not allow computations to be performed, such as combining a likelihood with a prior to calculate a posterior in accordance with Bayes theorem, or combining different sources of sensory evidence in a statistically optimal way, or computing functions of variables whose values are defined in probabilistic terms. However, as will be seen in the Results section, it is possible for the PC/BC-DIM algorithm to perform all of these forms of probabilistic inference. This is because PC/BC-DIM networks can be wired-up so that rather than representing the input probability distribution (the likelihood) the reconstruction neurons represent the posterior. In the simple case, where the PC/BC-DIM network reconstructs the input distribution, the reconstruction neurons can be interpreted as representing the posterior when the prior is uniform, and hence, the posterior is proportional to the likelihood. In order to calculate the posterior with a non-uniform prior, the prediction neuron RFs are scaled differently (see section 3.2). This results in prediction neurons with large weights (representing causes with a high prior probability) being preferentially selected by the PC/BC-DIM algorithm to represent the input distribution. As the posterior is represented by the

reconstruction neurons and is calculated as a combination of active prediction neuron RFs, the posterior will reflect the prior. When combining two sources of sensory evidence (see section 3.3), the prediction neurons have RFs that receive input from both sources. When there is a small cue conflict, the prediction neurons that have RFs which overlap with both input distributions are made active by the PC/BC-DIM algorithm. These prediction neurons produce a probability distribution at the reconstruction neurons that is intermediate between the two input distributions. Finally, to perform function approximation (see section 3.5), a PC/BC-DIM network is wired-up so that each prediction neuron has RFs representing the values of multiple variables. When probability distributions representing the likelihoods of a subset of variables are presented to the network, those prediction neurons whose RFs most closely match the given inputs are activated. Because these active prediction neurons have RFs tuned to the missing inputs, they also reconstruct a probability distribution representing the values of these missing inputs.

When multiple population codes are used as input to a PC/BC-DIM processing stage, it is convenient to think of the input vector being partitioned into separate sub-vectors representing the separate population codes, as illustrated in Fig. 1b. Each partition of the input encodes the probability distribution for a different variable. If the input is partitioned into multiple population codes, then the reconstruction neurons also represent multiple population codes, as illustrated in Fig. 1b. In all the experiments described in section 3, one or multiple probability distributions (represented as population codes) are provided as input to the PC/BC-DIM neural network. Each probability distribution is encoded as follows. Imagine a probability distribution $p(s|\omega)$ for a variable s . This is a continuous function over all possible values for s . This function can be encoded as a population code by sampling this continuous function at a finite number of locations, *i.e.*, at specific values of s . In all the simulations reported here the sampling locations were equally distributed.

2.4 Decoding and Quantitative Assessment Methods

Once a probability distribution has been represented by a population of neurons this can be used by the brain, and by the PC/BC-DIM model, as input to further probabilistic computations. Hence, the activity of the reconstruction neuron population does not need to be decoded. However, decoding is useful to demonstrate the accuracy of the model. The mean and variance are sufficient to fully characterise Gaussian probability distributions. To obtain these parameters the standard equations for calculating the mean (μ) and variance (σ^2) of a discrete probability distribution were used:

$$\mu = \frac{\sum_i z_i s_i}{\sum_i z_i} \quad (4)$$

$$\sigma^2 = \frac{\sum_i z_i (s_i - \mu)^2}{\sum_i z_i} \quad (5)$$

Where z_i is the activation of neuron i , and s_i is the RF centre (the preferred stimulus value) of neuron i . The denominator is necessary to normalise the reconstruction neuron responses (which can have arbitrary scaling) to form a valid probability distribution. Equation 4 is equivalent to the standard method of population vector decoding proposed by Georgopoulos et al. (1986).

In section 3.6, the calculations were performed with a variable (orientation) that wraps around. In this case the mean (in degrees) was calculated as:

$$\mu = \frac{180}{2\pi} \text{phase} \left(\frac{\sum_i z_i \exp \left(s_i \frac{2\pi\sqrt{-1}}{180} \right)}{\sum_i z_i} \right) \quad (6)$$

The experiments preceding section 3.6 can also be performed using variables with periodic boundary conditions with negligible effects on the results.

To decode the reconstruction neuron responses, the vector of neural activations, \mathbf{z} , was set equal to \mathbf{r}^c in the above equations. Where c is an integer equal to the number of cues to the same sensory stimulus. Hence, in experiments where cue integration or segregation was performed with two cues (Figs. 6, 8, 9, 11c-d, 14d-g, and 15) the squared response of the reconstruction neurons was used for decoding, *i.e.*, $\mathbf{z} = \mathbf{r}^2$ was used in equations 4–6. In the experiment on cue integration with three cues (Fig. 7) the cubed reconstruction neuron responses were used, *i.e.*, $\mathbf{z} = \mathbf{r}^3$. In all other cases $\mathbf{z} = \mathbf{r}$ was used.

Using exponential responses to decode the posterior in cue integration tasks is necessary in order to produce an accurate estimate of the variance of the posterior distribution in those tasks (otherwise the variance would be over-estimated). In the brain, downstream neurons performing further probabilistic computations would need to know c in order to be able to raise the reconstruction neuron responses to the correct power. It is fairly easy to imagine additional neural circuitry that could calculate c , but the need to raise the reconstruction neuron responses to different powers for different computations is a limitation of the current model. However, it is a fairly minor

limitation compared to previous work which has either proposed completely different algorithms for performing cue integration (Ma et al., 2006) and function approximation (Beck et al., 2011), or which has failed to compute the variance of the posterior at all (Deneve et al., 2001; Pouget et al., 2003).

2.5 Code

Open-source software, written in MATLAB, which performs the experiments described in this article is available from: http://www.corinet.org/mike/Code/pcbc_prob.zip.

3 Results

The results of example simulations are presented in a standard format like that shown in Fig. 2b. In these figures the lower histogram shows the input to the PC/BC-DIM network which is a population code describing the input probability distribution (or likelihood), $p(s|\omega)$. The length of each bar (indicated on the y-axis) represents the probability $p(s_i|\omega)$ at a specific value of the variable s (indicated by the labels on the x-axis). The middle histogram shows the responses of the prediction neurons. The y-axis is in arbitrary units representing firing rate and the x-axis is labelled with neuron number. The upper histogram shows the responses of the reconstruction neurons. The length of each bar (indicated on the y-axis) represents the firing rate of the neuron, and hence, the value of the posterior probability distribution at a specific value of s (indicated by the labels on the x-axis). In most simulations the values of s are measured in units of degrees. This is simply to give these values concrete units and does not mean that PC/BC-DIM is limited to computing with variables measured in degrees. For each experiment the weights of the PC/BC-DIM network have been set by trial and error to produce good results on that task.

3.1 Decoding Noisy Population Coded Probability Distributions

One important issue when dealing with population codes that encode probability distributions is how to deal with random fluctuations in the neural firing rates: how to accurately estimate the probability distribution despite the samples being unreliable due to corruption by noise. A decoding method is thus required that can combine noisy input samples to calculate an estimate of the underlying probability distribution. Previous work has shown that, when the probability distribution is Gaussian, attractor neural networks can be used to convert a noisy population code into a smooth Gaussian centred close to the maximum likelihood value (Deneve et al., 1999; Latham et al., 2003; Pouget et al., 2000, 1998). Such decoding would allow the statistically optimum estimate to be easily read off as the peak of the output distribution.

The PC/BC-DIM network can also perform near optimum decoding of a noisy, Gaussian, probability distribution. While PC/BC-DIM is not limited to dealing with Gaussian probability distributions, we consider 1D Gaussian probability distributions as this allows direct comparison with previous work as well as with the results that would be expected from exact Bayesian inference. The prediction neurons are given Gaussian RFs (all with standard deviation 10°) covering the range of possible values (means distributed uniformly in the range -180° to 180°), as shown in Fig. 2a. The PC/BC-DIM network reconstructs the input as a linear combination of basis functions (the prediction neuron RFs, see Methods). In this case, the input probability distribution is reconstructed as a combination of Gaussians. This fitting of a set of Gaussians to the data can be seen as a form of kernel density estimation of the input distribution. When the input is a noisy Gaussian population code, the reconstruction is a smooth Gaussian, as shown for two specific examples in Fig. 2b and c. The smoothing effect results from each Gaussian RF receiving input from a number of samples of the input distribution, which means that noise is averaged out. The accurate reconstruction of the input distribution results from the PC/BC-DIM algorithm minimising the KL divergence between the reconstruction and the input. To confirm that the network’s estimate of the probability distribution is close to the statistically optimal estimate, in general, experiments were performed using Gaussian input distributions with random mean values chosen uniformly from the range $[-90^\circ : 90^\circ]$ and random standard deviations chosen uniformly from the range $[15^\circ : 45^\circ]$. Each input distribution was corrupted using poisson noise which is commonly used to simulate noise in biological neurons. To do so, each input activation was a sample taken from a poisson distribution whose mean was the noise-free value of that input. Figs. 2d and e show plots of the network’s estimate of the mean and variance of the probability distribution (given by the reconstruction neurons) compared to the optimal estimates of these parameters (calculated from the input distribution) for 100 trials. It can be seen that both the mean and variance are very accurately estimated by the PC/BC-DIM network in all 100 trials. To quantify the decoding accuracy the absolute difference between the mean of the probability distribution given by PC/BC-DIM and the statistically optimal value was calculated for each of 100 000 trials. The maximum absolute difference was 0.36° , the median absolute difference was 0.002° and the mean absolute

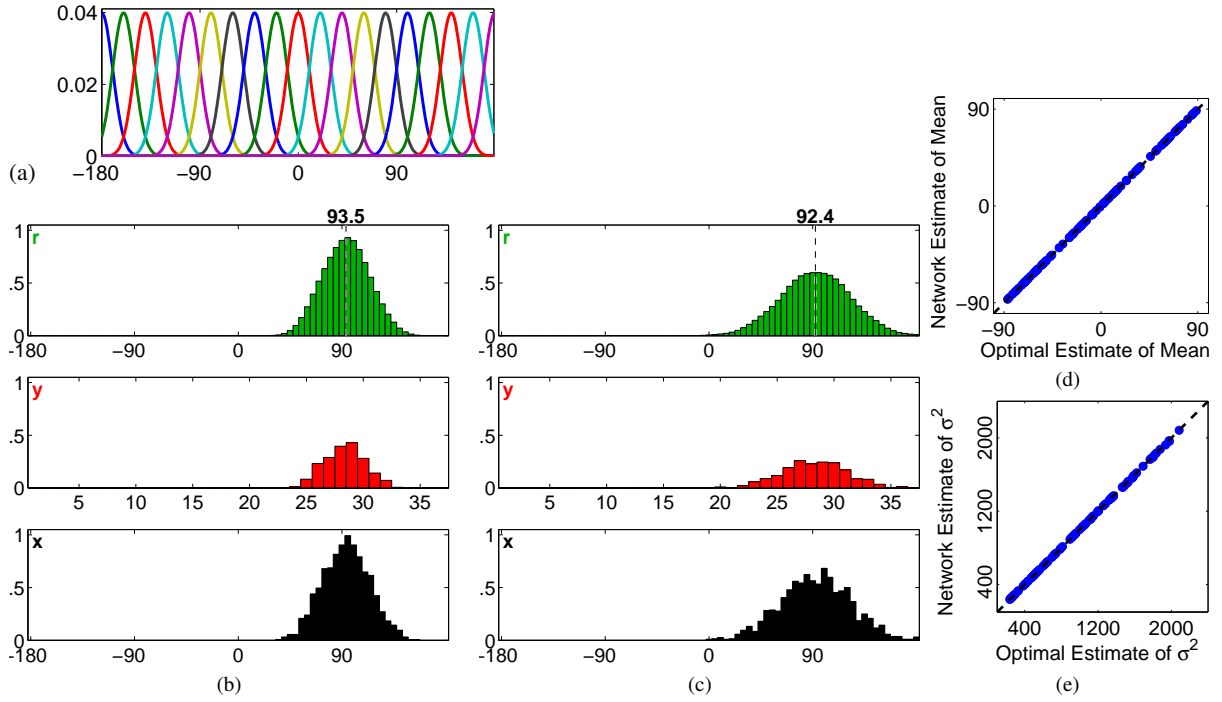


Figure 2: Decoding noisy population coded probability distributions. (a) The uniform population of Gaussian RFs used for the \mathbf{W} weights in the simulations reported in (b)-(e), and Fig. 4. Note that for clarity the RFs of every other neuron are shown at a finer sampling rate (1°) than has been used to define the weights used in the simulations (5°). (b) and (c) Example simulation results. Each example shows an input population code representing a Gaussian probability distribution (bottom histograms) with mean 93° that has been corrupted by Poisson noise. The standard deviations of the original distributions are (a) 20° , and (b) 30° . The middle histograms show the prediction neuron activations, and the upper histograms show the reconstruction neuron responses. The numbers above each histogram show the maximum likelihood estimate of the stimulus value calculated from that histogram (equation 4). Note that if the input population code was normalised to sum to unity this would result in the population code generated by the reconstruction neurons also summing to one. (d) Each dot shows a comparison between the PC/BC-DIM network’s estimate of the mean of the probability distribution (given by the reconstruction neurons) and the statistically optimal estimate of the mean of the input distribution. 100 experiments were performed using input distributions with randomly chosen means and standard deviations that were encoded with noisy population codes. (e) As for (d) but for variance.

difference was 0.014° . The percentage absolute difference between the network’s estimate of the variance and the statistically optimal value, over the same 100 000 trials, had a maximum of 1.8%, a median of 0.10%, and a mean of 0.18%.

Note that the network has been used to estimate the parameters of a single noisy probability distribution (*i.e.*, the same corrupted input was presented to the network during all iterations of the PC/BC-DIM algorithm). A more biologically valid approach would model the variability of neural responses in the inputs and within the PC/BC-DIM network. Doing this leads to a reduction in the accuracy in the estimate of the posterior probability distribution. Specifically, the median absolute difference between the network’s estimate of the mean and the optimal estimate of the mean drops to 2.46° if the noise on the input changes each iteration, and is 2.80° if, additionally, Poisson noise is added to the prediction neuron responses at each iteration (the corresponding errors in the estimates of the variance are 11.6% and 55.3%). To improve the performance in these circumstances it would be necessary to modify the PC/BC-DIM algorithm to update neural responses slowly, and hence, to estimate mean firing rates.

Zemel et al. (1998) performed an experiment to show that the KDE model (Anderson and Van Essen, 1994; Barber et al., 2003) is incapable of reconstructing probability distributions which are narrower than the basis functions. In common with the KDE model, PC/BC-DIM also reconstructs the probability distribution as a linear sum of basis functions, and thus, it also suffers from this limitation. To demonstrate this, PC/BC-DIM was tested using the experiment described in Zemel et al. (1998). In this experiment, the prediction neurons had Gaussian RFs

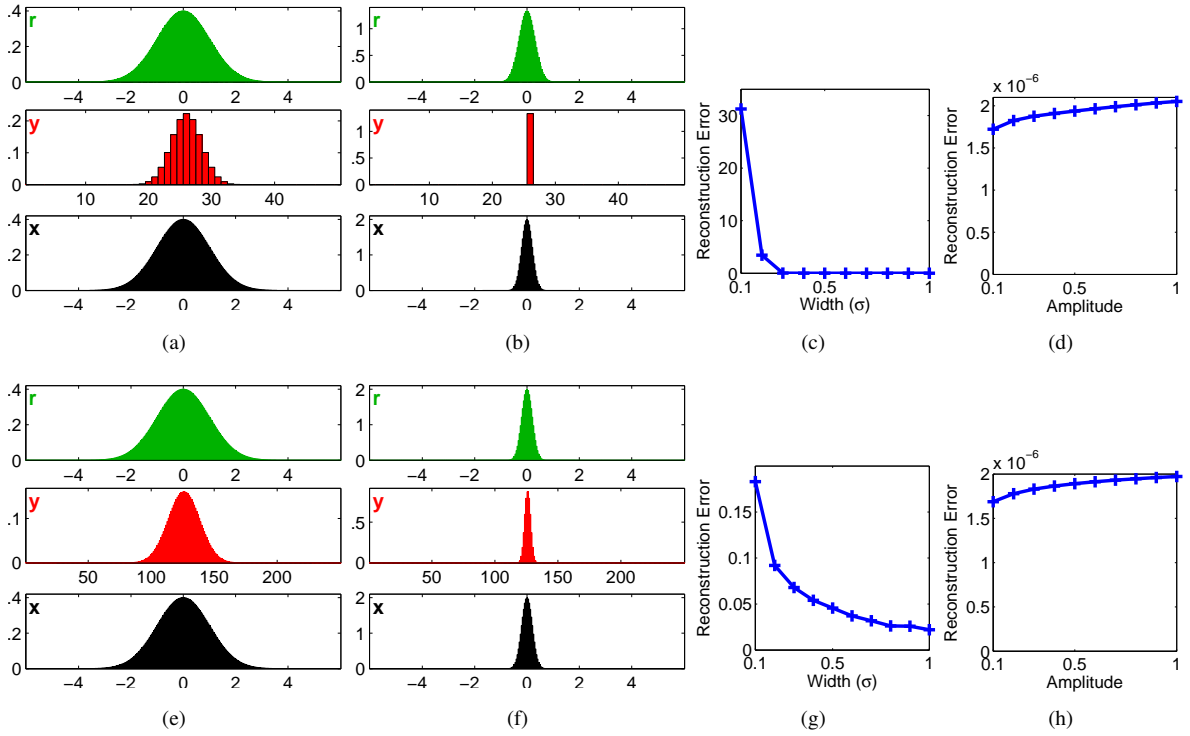


Figure 3: Effects of width and height when decoding population coded probability distributions. Results are for prediction neurons with Gaussian RFs that have a standard deviation of 0.3 (top row), and 0.08 (bottom row). (a) and (e) the input distribution has a standard deviation of 1. (b) and (f) the input distribution has a standard deviation of 0.2. (c) and (g) The sum of the squared difference between x and r as a function of the width of the input distribution. (d) and (h) The sum of the squared difference between x and r as a function of the amplitude of the input distribution.

with a standard deviation of 0.3 uniformly spaced in the range $[-10 : 10]$. As in the preceding experiments, when the input distribution was wider than the RFs the reconstruction was accurate (Fig. 3a). However, when the input distribution was narrower than the prediction neuron RFs the distribution encoded by the reconstruction neurons was too wide (Fig. 3b). To quantify this effect, Zemel et al. (1998) performed experiments with noisy population codes of different widths and calculated the sum of the squared difference between the reconstruction and the true (uncorrupted) distribution. For the PC/BC-DIM algorithm, this reconstruction error was large for narrow input distributions (Fig. 3c), due to the inability of the PC/BC-DIM network to accurately represent distributions narrower than the prediction neuron RFs. The prediction neuron weights are basis functions or elementary components that can be used to reconstruct a probability distribution. Clearly, these components need to be appropriate for a given task, or the reconstruction will be poor. Hence, to represent narrow distributions a network would need prediction neurons with narrow RFs. Repeating the preceding experiment using RFs with a standard deviation of 0.08 did result in more accurate results (Fig. 3e-g). However, this is at odds with biological data which shows that discrimination is more finely tuned than the RFs of cortical neurons (Zemel et al., 1998).

Another issue explored in Zemel et al. (1998) is the inability of the KDE model to accurately represent probability distributions with small amplitude. To assess this Zemel et al. (1998) performed experiments with Gaussian input distributions of varying height, and they calculated the sum of the squared difference between the input distribution and the one reconstructed by the KDE algorithm. Results for the same experiments performed with PC/BC-DIM are shown in Figs 3d and h. It can be seen from these results that the PC/BC-DIM algorithm is capable of very accurately representing probability distributions regardless of their amplitude.

Some previous neural models of decoding (Deneve et al., 1999; Latham et al., 2003; Pouget et al., 2000, 1998) have used attractor networks that can only generate a single mono-modal Gaussian distribution, and hence, can not simultaneously represent multiple, distinct, stimuli (Sahani and Dayan, 2003). Unlike these previous algorithms, PC/BC-DIM is not limited to representing a mono-modal distribution, as illustrated in Fig. 4. When encoding probability distributions using population codes there is an inherent ambiguity between a complex distribution that represents uncertainty about a single cause, and a complex distribution that represents multiple separate causes (Sahani and Dayan, 2003). This work does not address this issue, but multi-modal distributions are treated

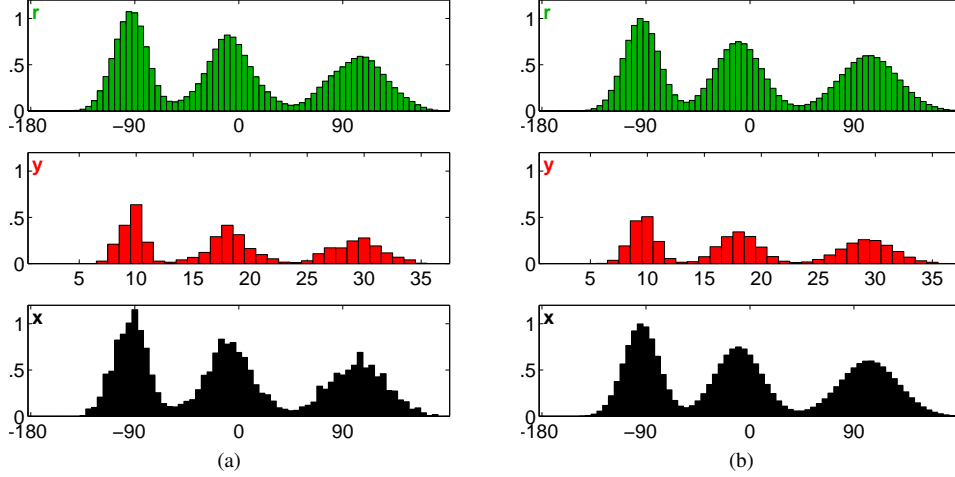


Figure 4: Decoding multi-modal population coded probability distributions. Each example shows a tri-modal input distribution (bottom histograms) with poisson noise (a), and without noise (b). The middle histograms show the prediction neuron activations, and the upper histograms show the reconstruction neuron responses. The network was identical to that used to produce the results shown in Fig 2.

throughout as representations of multiple stimuli.

3.2 Combining Likelihoods and Priors to Calculate Posterior Probability Distributions

The defining characteristic of Bayesian inference is that it makes use of prior information. Specifically, Bayes theorem expresses how the likelihood (the probability distribution derived from the current observation) should be combined with the prior (the probability derived from our knowledge about the state of nature) in order to calculate the posterior distribution. Surprisingly, many previous neural implementations of Bayesian inference with population codes have ignored priors, or equivalently have assumed that the prior is flat, and have hence failed to demonstrate an ability to perform Bayesian inference (Deneve et al., 1999, 2001; Jazayeri and Movshon, 2006; Latham et al., 2003; Pouget et al., 2013, 2003, 2000, 1998; Seilheimer et al., 2014; Zemel et al., 1998). Other theories propose that priors are represented by spontaneous activity (Fiser et al., 2010) or by a separate population of neurons whose activity is added to the activity of the population of neurons representing the likelihood (Ma et al., 2006). Representing the prior using the activity of a separate population of neurons would also enable the prior to be combined with the likelihood in the same way that two sensory cues can be integrated (see section 3.3). Alternatively, it has been proposed that the prior can be encoded by the distribution of RFs, such that more neurons, typically with narrower tuning widths, are allocated to representing stimulus values that are more probable (Ganguli and Simoncelli, 2010, 2014; Girshick et al., 2011; Shi and Griffiths, 2009). Storing priors in the synaptic weights makes intuitive sense as priors result from previous experience and should change relatively slowly (Vilares and Kording, 2011). The current model also incorporates the prior into the synaptic weights of the network, however, this is done by scaling the weights of uniformly distributed RFs, rather than by changing the distribution of the RFs.

Figure 5 shows a specific example of how PC/BC-DIM succeeds in calculating a posterior distribution (represented by the reconstruction neuron activations) by combining a likelihood (represented by the input population code) and a prior (incorporated in the synaptic weights of the network). The network is identical to that used to generate the results shown in Fig. 2, except that in this experiment the weights have been changed as shown in Fig. 5a. The prior probability distribution is a Gaussian centred at 0° and with a standard deviation of 60° . Each neuron’s weights (the rows of \mathbf{W}) have simply been multiplied by this prior distribution (\mathbf{V} is set equal to the transpose of \mathbf{W} and re-normalised so the each column has a maximum value of one, as in all other experiments). For the two examples shown in Fig. 5b and c the firing rates of the reconstruction neurons provide an almost exact approximation to the posterior that would be calculated via Bayes theorem. For an intuitive understanding of why this happens consider the result in Fig. 5c. In this example, the two prediction neurons with the highest responses have RFs centred at 80° and 70° . These RFs are less similar to the input distribution than the neuron with an RF centred at 90° . However, the neurons with RFs centred at 80° and 70° have weights that are larger in magnitude. The product of the weight vector and the input population code is thus greater for the prediction neurons with RFs

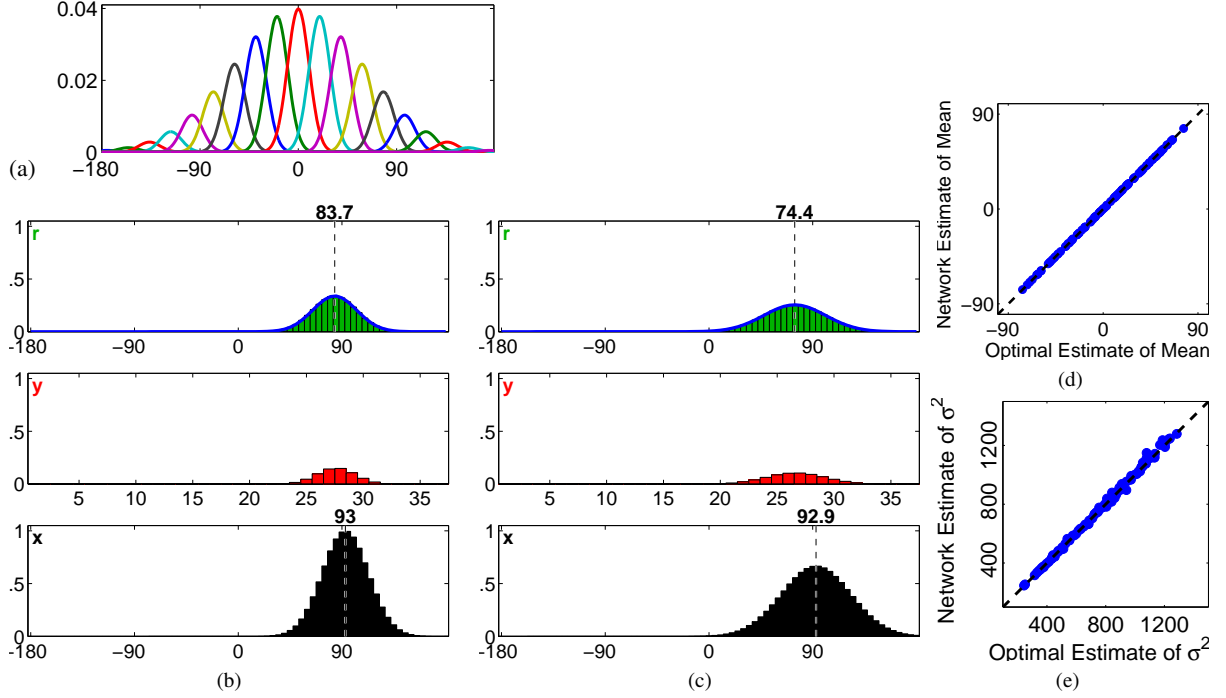


Figure 5: Bayesian inference with a prior. (a) RFs incorporating a prior as used for the W weights in the simulations reported in (b)-(e). Note that for clarity the RFs of every other neuron are shown at a finer sampling rate (1°) than has been used to define the weights used in the simulations (5°). (b) and (c) Example simulation results. Each example shows a Gaussian input distribution representing the likelihood (bottom histograms), and the response of the reconstruction neurons which represent the posterior (upper histograms). The numbers above each histogram show the maximum likelihood estimate of the stimulus value calculated from that histogram (equation 4). Due to the prior being centred at 0° the posterior probability distributions in these experiments are shifted towards 0° compared to the corresponding experiments shown in Fig. 2. The Gaussian curves superimposed on the upper histograms show the posterior distribution calculated via exact Bayesian inference (scaled in amplitude to fit the network's estimate). (d) Each dot shows a comparison between the PC/BC-DIM network's estimate of the mean of the posterior probability distribution (given by the reconstruction neurons) and the mean of the posterior calculated via exact Bayesian inference. 100 experiments were performed using input distributions with randomly chosen means and standard deviations that were encoded with noisy population codes. (e) As for (d) but for variance.

centred at 80° and 70° than it is for the prediction neuron with an RF centred at 90° . This results in the higher firing rates of these two neurons. As the posterior (encoded by the reconstruction neuron responses) is a linear combination of basis functions (RFs) weighted by the corresponding prediction neuron firing rates, the posterior peaks between 80° and 70° , and is thus shifted towards the prior.

For clarity, Figs. 5a and b show results for calculations performed using likelihoods that have not been corrupted with noise. However, PC/BC-DIM performs almost exact Bayesian inference even when the input distributions are noisy. To confirm that the network's estimate of the posterior probability distribution is close to the statistically optimum estimate, 100 000 trials were performed using likelihood distributions corrupted using poisson noise. In each trial the likelihood was given a random mean, chosen uniformly from the range $[-90^\circ : 90^\circ]$, and a random standard deviation chosen uniformly from the range $[15^\circ : 45^\circ]$. Over 100 000 trials, the absolute difference between the optimal estimate of the mean of the posterior distribution (calculated by Bayes theorem) and the network estimate of the mean (from the PC/BC-DIM reconstruction neuron responses) had a maximum value of 1.23° . The median absolute difference was 0.07° , and the mean absolute difference was 0.11° . Over the same 100 000 trials, the variance of the posterior distribution given by the network and by Bayes theorem had a maximum, median, and mean, percentage absolute difference of 15.9%, 0.98%, and 1.35% respectively. Figs. 5d and e show plots of the network's estimate of the mean and variance of the probability distribution compared to the optimal estimates of these parameters for 100 trials.

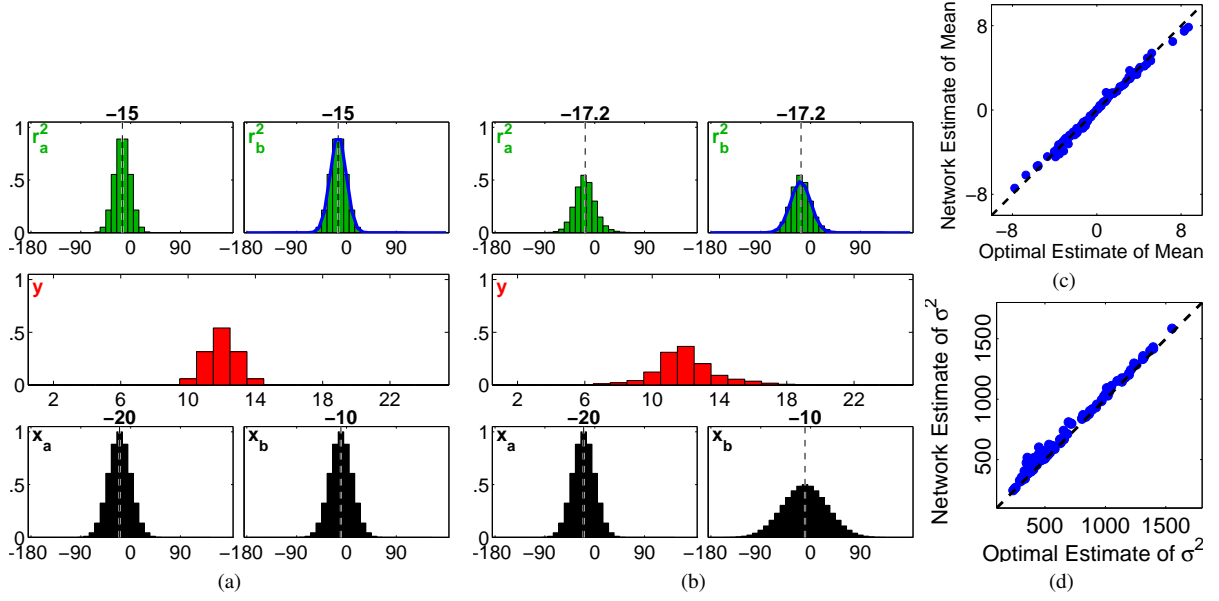


Figure 6: Cue integration. (a) and (b) Example simulation results. Each example shows two Gaussian input distributions (bottom histograms), representing two cues to the same sensory stimulus. The network produces, in the reconstruction neuron activations (upper histograms), probability distributions that closely match those that would be obtained by the statistically optimal combination of the two cues. The Gaussian curves superimposed on the upper-right histograms show the posterior distribution calculated via exact Bayesian inference (scaled in amplitude to fit, approximately, the network’s estimate). The numbers above each histogram show the maximum likelihood estimate of the stimulus value calculated from that histogram (equation 4). (c) and (d) Show cue integration accuracy for noisy population coded probability distributions. (c) Each dot shows a comparison of the PC/BC-DIM network’s estimate of the mean of the combined probability distribution compared to the probabilistically optimal estimate of the mean. 100 experiments were performed with randomly chosen cue conflicts and cue precisions. As in [Ma et al. \(2006\)](#), each dot represents the average of 1008 trials performed with different noisy population codes. (d) As for (c) but for variance.

3.3 Cue Integration

In many circumstances multiple sources of information may be available about the same sensory stimulus. Cue integration results in these separate sources of sensory evidence being combined together to produce a single estimate of the stimulus’ properties. The distinct cues may be derived from the same sensory modality, such as when estimating depth from multiple visual cues like disparity and linear perspective, or may come from different modalities, such as when estimating depth using vision and proprioception. Various experiments have shown that human performance in cue integration tasks is optimal, which requires the reliability of each sensory cue to be taken into account ([Seilheimer et al., 2014](#)). For cues that can be represented by Gaussian probability distributions, and assuming a flat prior, the mean of the combined estimate is the sum of the means of the two cues weighted by the precision (the inverse of the variance) of each cue ([Ernst and Jäkel, 2003](#); [Ma et al., 2006](#); [Ma and Pouget, 2008](#); [Pouget et al., 2013](#)).

Figure 6 illustrates that optimal cue integration can be performed using PC/BC-DIM. In these experiments the input was partitioned into two in order to represent the two cues. For convenience both cues were measured in the same units and could take values over the same range, but this is not a requirement. Each prediction neuron had a Gaussian RF (with standard deviation 15°) centred at the same location in each input space. The population of predictions neurons had RFs covering the range of possible values (means distributed uniformly in the range -180° to 180°). When population codes representing Gaussian probability distributions were presented to the two input spaces, the PC/BC-DIM network generated population codes (the reconstruction neuron responses) that peaked very near to the optimal estimate obtained by probabilistically combining the two cues. Because there are two partitions of the input, there are also two partitions of the reconstruction neuron populations. Both represent the combined estimate of the stimulus values, and hence, both generate the same population code. For an intuitive understanding of how PC/BC-DIM performs cue integration consider the result shown in Fig. 6a. Here,

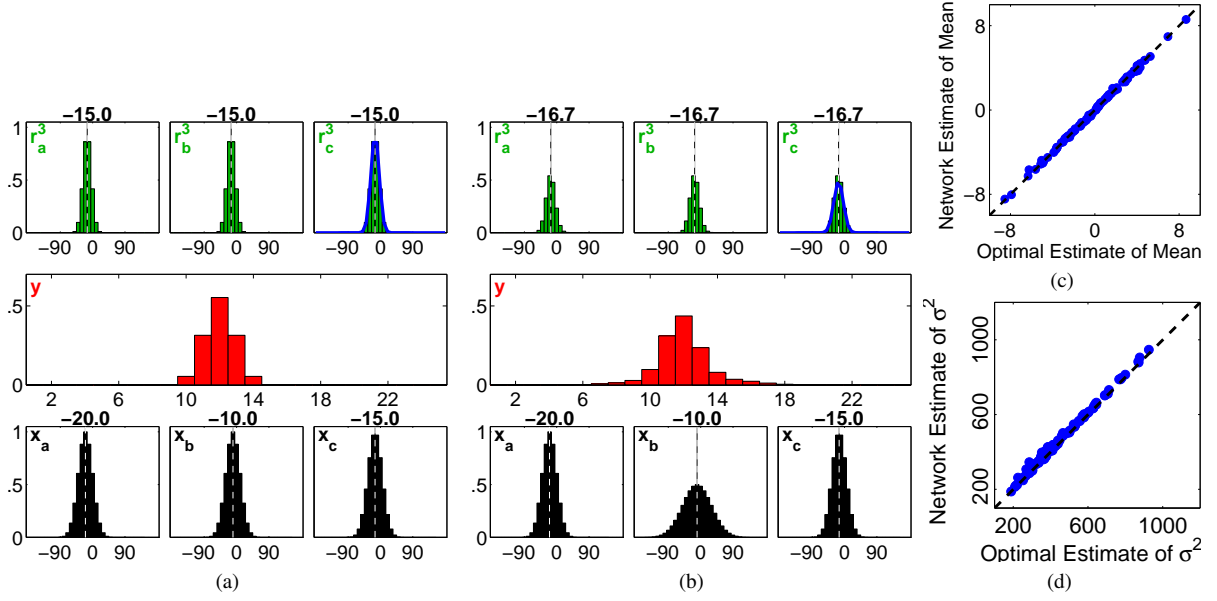


Figure 7: Cue integration with three cues. The format of this figure is identical to, and described in the caption of, Fig. 6, except that here there are three Gaussian input distributions (bottom histograms), representing three cues to the same sensory stimulus.

the most responsive prediction neuron has a RF centred at -15° in both input spaces. This is the most active prediction neuron because both its RFs overlap with the input distributions, it thus receives the most support. The reconstruction neuron responses are a linear combination of all the active prediction neuron RFs, and hence, also peaks at -15° in both input spaces.

For clarity, Figs 6a and b show results for calculations performed using input population codes that have not been corrupted with noise. However, PC/BC-DIM performs near optimal cue integration even when the input distributions are noisy. To confirm this the method used in Ma et al. (2006) was employed. The two Gaussian input distributions were both corrupted by poisson noise. One distribution had a fixed mean (0°) while the other had a randomly chosen mean (uniformly selected from the range $[-12^\circ : 12^\circ]$), so that cue conflict varied by up to $\pm 12^\circ$. Each input distribution was assigned a random standard deviation, chosen uniformly from the range $[20^\circ : 60^\circ]$. Figs. 6c and d show plots of the network's estimate of the mean and variance of the combined probability distribution compared to the optimal estimates of these parameters. To quantify these results, the maximum, median, and mean absolute difference between the network's and the optimal estimate of the mean was 0.76° , 0.18° , and 0.24° . The maximum, median, and mean percentage absolute difference between the network's and the optimal estimate of the variance was 34.62%, 5.03%, and 7.26%.

Repeating the above experiment but for cue integration with three sensory cues produced the results shown in Fig 7. An identical network was used except that each prediction neuron received input from three partitions representing the three cues. Again, the network's estimate of the combined probability distribution was accurate. Specifically, the maximum, median, and mean absolute difference between the network's and the optimal estimate of the mean was 0.58° , 0.07° , and 0.09° . The maximum, median, and mean percentage absolute difference between the network's and the optimal estimate of the variance was 20.60%, 2.69%, and 3.72%.

PC/BC-DIM can also perform optimal cue integration with a non-flat prior. As in section 3.2, the prior was incorporated by modifying the relative strengths of the synaptic weights. Figures 8a and b show illustrative results when the prior probability distribution, for both cues, was a Gaussian centred at 0° and with a standard deviation of 60° . Each neuron's weight vector, in both input spaces, was multiplied by this prior distribution. To demonstrate that the network's cue integration with a prior is near optimal even when the input distributions are noisy, the experiment described in the previous paragraphs was repeated, and the results are shown in Figs. 8c and d. The maximum, median, and mean absolute difference between the network's and the optimal estimate of the mean was 0.28° , 0.05° , and 0.06° . The maximum, median, and mean percentage absolute difference between the network's and the optimal estimate of the variance was 10.11%, 1.59%, and 2.35%.

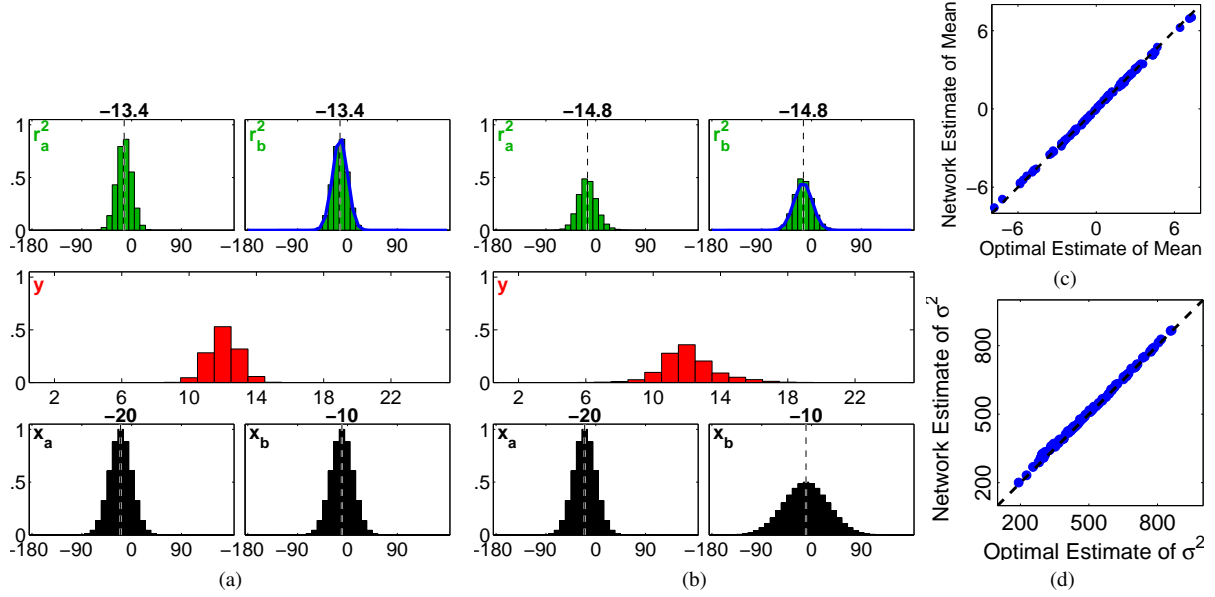


Figure 8: Cue integration in the presence of a prior. (a) and (b) Example simulation results. (c) and (d) Show cue integration accuracy for noisy population coded probability distributions. The format of this figure is identical to, and explained in the caption of, Fig. 6. Due to the prior being centred at 0° the posterior probability distributions in these experiments (represented by the reconstruction neuron responses in the upper histograms in (a) and (b)) are shifted towards 0° compared to the corresponding experiments shown in Fig. 6. The Gaussian curves superimposed on the upper-right histograms show the posterior distribution calculated via exact Bayesian inference (scaled in amplitude to approximately fit the network’s estimate).

3.4 Cue Segregation

In multisensory integration experiments with human subjects, large cue conflict results in cue segregation rather than integration (Beierholm et al., 2008). In which case the two cues are not perceived to have the same cause. Determining if two cues should be integrated or segregated (*i.e.*, determining if cues arise from a single cause, or from multiple, independent causes) can be posed in terms of an inference problem: causal inference (Ma and Pouget, 2008; Seilheimer et al., 2014; Shams and Beierholm, 2010; Vilares and Kording, 2011). While there exist Bayesian models of causal inference, there are no current neurally-based models.

Existing neural models of cue integration and the closely related task of function approximation (see section 3.5) either take the form of (radial) basis function networks (Deneve and Pouget, 2003; Pouget and Sejnowski, 1994, 1997; Pouget and Snyder, 2000; Salinas and Abbott, 1995; Salinas and Sejnowski, 2001) or attractor networks (Deneve et al., 1999; Latham et al., 2003; Ma et al., 2006; Pouget et al., 2002, 1998). Both types of model require that all the sensory information originates from a single source. Hence, if sensory inputs originating from multiple underlying causes are presented simultaneously, these networks will erroneously combine this information into a single, incorrect, estimate (Pouget et al., 2002). Thus, while such networks can simulate cue integration, they are incapable of modelling cue segregation. In contrast, PC/BC-DIM can model cue segregation as well as cue integration. Unlike the quantitative analysis of cue integration in the preceding section, the assessment of cue segregation offered here is only qualitative. Specifically, segregation is assumed to have occurred if the probability distribution represented by the reconstruction neurons is multi-modal: as mentioned in section 3.1, multi-modal distributions are treated as representations of multiple stimuli.

Figure 9a shows that when two inputs encode very different stimulus values the reconstruction neurons generate a bi-modal probability distribution with peaks at the positions of the two cues. This is in contrast to when there is less cue conflict, as in Fig 6a, where the reconstruction neurons generate a mono-modal distribution that peaks at the weighted mean of the two cues. PC/BC-DIM can perform cue segregation and integration simultaneously, as illustrated in Fig. 9b. Here, one input space receives a bi-modal population code that peaks at -30° and $+50^\circ$, and the other input space receives a mono-modal distribution with a mean of $+60^\circ$. The reconstruction neurons generate a bi-modal distribution with peaks at approximately -30° and $+55^\circ$. The first peak is thus the result of information presented to only one input space (*i.e.*, cue segregation), while the second peak is an estimate of the stimulus value based on combining two cues that have the same precision presented to different input spaces

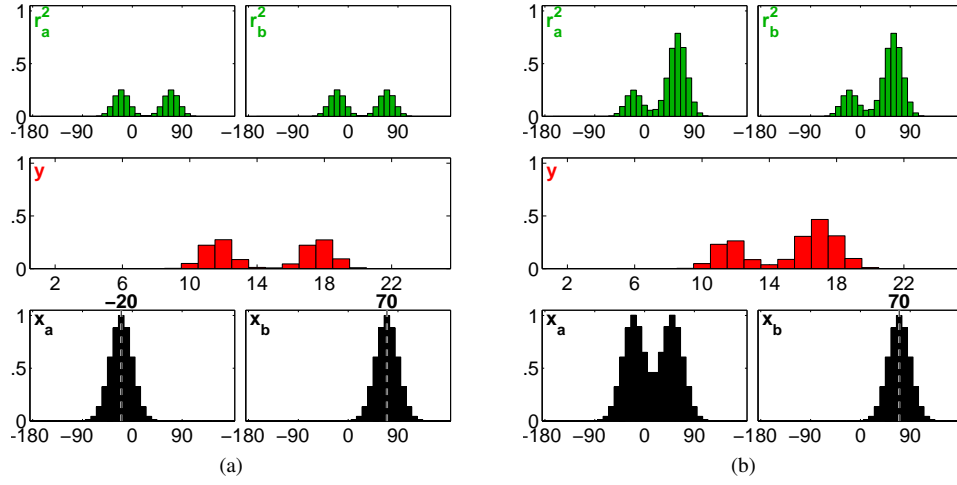


Figure 9: Cue segregation. (a) Shows two Gaussian input distributions (bottom histograms), representing two cues with a large cue conflict. The network produces, in the reconstruction neuron responses (upper histograms), bi-modal probability distributions representing stimuli at both locations predicted by the two cues. (b) As for (a) but with a second cause represented by the bi-model distribution presented to the first input space. This second cause is integrated with the cue presented to the other input space.

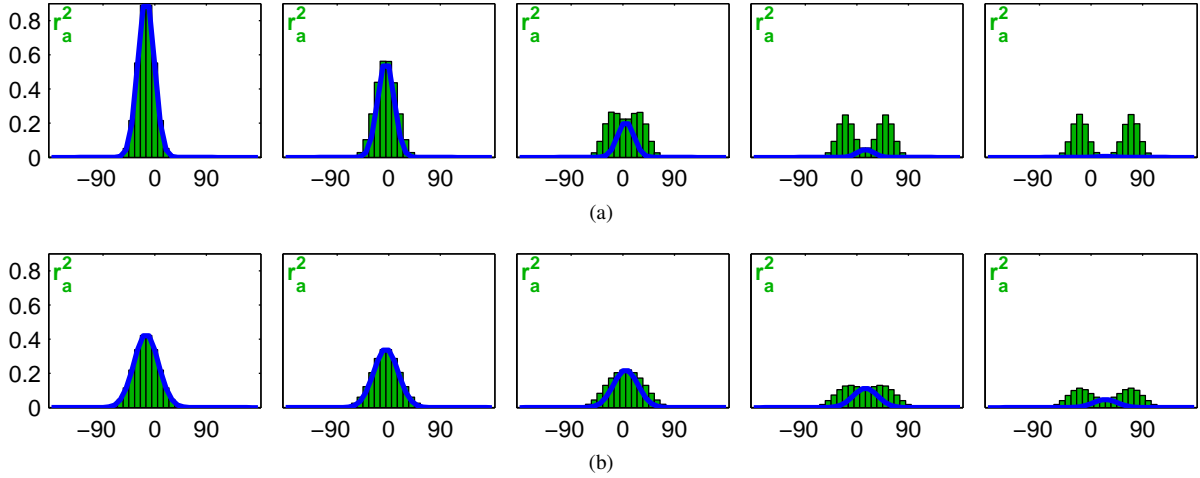


Figure 10: Causal inference. The input (not shown) consist of two cues encoded as Gaussian probability distributions with equal precisions. Cue conflict increases from left to right, from 10° to 90° in steps of 20° . Each subplot shows the responses of the first partition of the reconstruction neurons. The Gaussian curves superimposed on the histograms show the posterior distributions calculated via exact Bayesian inference for cue integration. (a) Each cue has a standard deviation of 20° . The left most subplot, showing results for a conflict of 10° , is a repeat of the experiment shown in Fig. 6a. The right most subplot, showing results for a conflict of 90° , is a repeat of the experiment shown in Fig. 9a. (b) As (a) except cues have a standard deviation of 30° . In both cases, as cue conflict increases the probability distribution generated by PC/BC-DIM changes from mono-modal (cue integration) to bi-modal (cue segregation). When cues have lower precision, as shown in (b), cue integration occurs for a wider range of cue conflicts.

(i.e., cue integration). In PC/BC-DIM whether signals are integrated or segregated depends on the degree of cue conflict and the precision of the two cues. For cues with small standard deviation, segregation will occur at a smaller conflict than when the cues have larger standard deviation, as illustrated in Fig. 10.

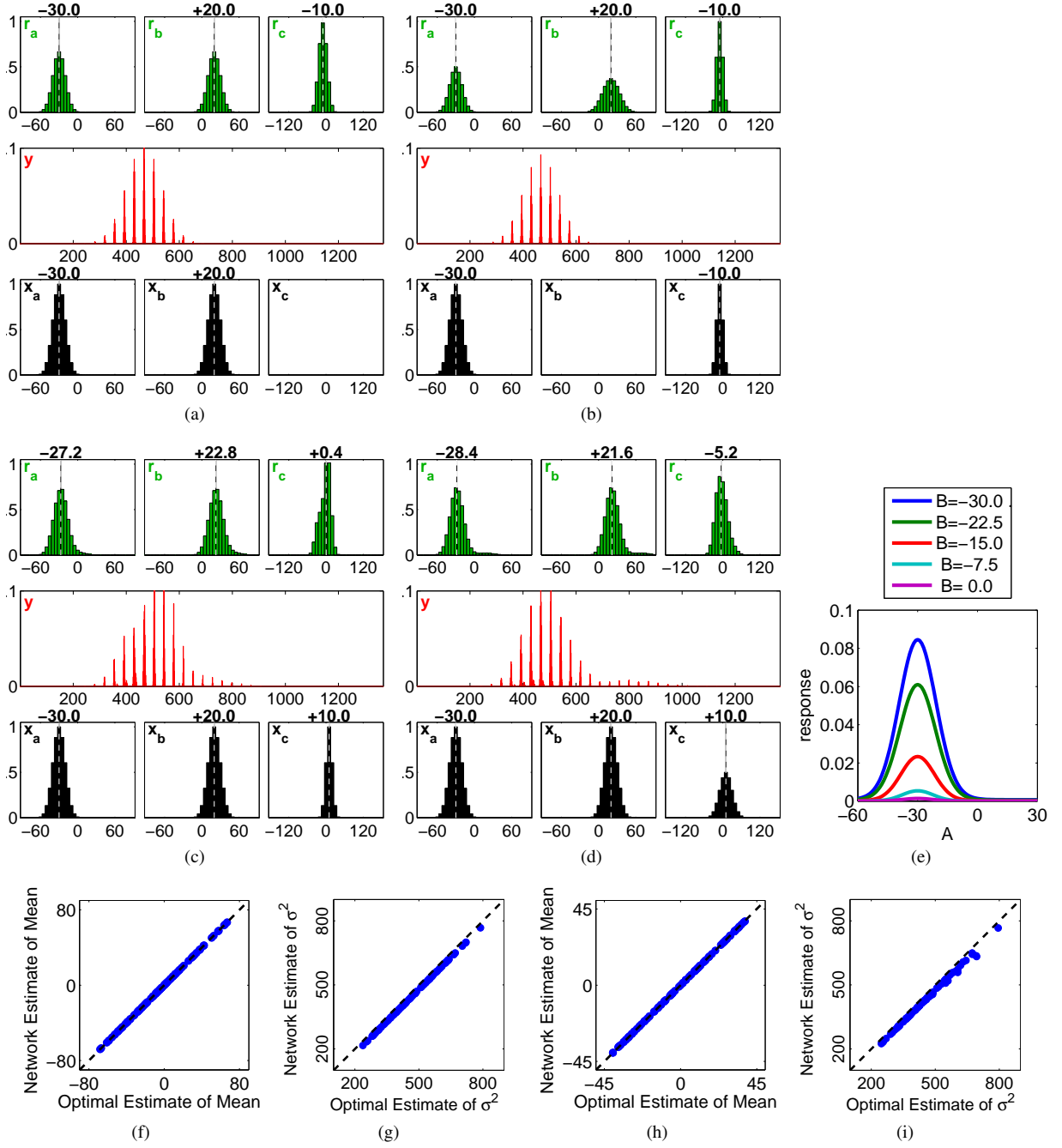
3.5 Function Approximation

Function approximation is required for many tasks faced by the brain. For example, to perform sensory-sensory coordinate transformations in order to bring different sources of sensory information into a common reference frame, or to perform sensory-motor mappings in order to control movement. Here, to allow direct comparison with previous work (Beck et al., 2011; Deneve et al., 2001; Pouget et al., 2002; Pouget and Sejnowski, 1997; Pouget and Snyder, 2000) we only consider simple, linear, functions of one-dimensional variables, encoded using Gaussian distributions. However, like this previous work, PC/BC-DIM is not limited to linear function approximation, nor is PC/BC-DIM limited to computing with one-dimensional Gaussian inputs as has been shown previously (De Meyer and Spratling, 2011, 2013).

Previous work (Beck et al., 2011; Deneve et al., 2001; Pouget et al., 2002; Pouget and Sejnowski, 1997; Pouget and Snyder, 2000) has considered approximating a function of three variables (A, B, and C) such that $C=A+B$. If, for example, A is considered to be a representation of the retinal position of an object, and B a representation of eye position, then C can be considered a representation of the head-centred bearing of the object. Given any two of these values, existing models can calculate the third (Deneve et al., 2001; Pouget et al., 2002). Alternatively, if supplied with all three values existing networks can perform cue integration (Deneve et al., 2001; Pouget et al., 2002). Furthermore, it has been shown that neurons in such networks display gain modulated responses, similar to those observed in the dorsal pathway of the cortex (Deneve et al., 2001; Pouget et al., 2002). PC/BC-DIM can reproduce all of these results, as is illustrated in Fig. 11a-e. For this task we can consider the input to the PC/BC-DIM network, and hence, the reconstruction produced by the network to be partitioned into three parts, as illustrated in Fig. 1b. Each partition represents a population coded probability distribution encoding the uncertainty about the value of a different variable (A, B, or C). Each prediction neuron has a Gaussian RF, of standard deviation 5° , in each of the three partitions. These RFs are centred in each input space so as to encode the relationship $A+B=C$ for one specific set of values. The RFs of the population as a whole evenly tile the A and B input spaces, so that the network can approximate $C=A+B$ for all values of A and B. For an intuitive understanding of how the PC/BC-DIM network performs function approximation, consider the result shown in Fig. 11a. The two input distributions cause responses in the subset of prediction neurons with RFs that are centred near -30° in the first partition and near 20° in the second partition. Each of these prediction neurons has an RF centred near -10° in the third partition. The reconstruction neuron responses are a linear combination of all the active prediction neuron RFs, and hence, will peak at the appropriate places in each of the three partitions.

To confirm that the PC/BC-DIM network can perform accurate function approximation even when the population codes are corrupted with noise, the method used in Deneve et al. (2001) was employed. Values for A and B were chosen at random (uniformly from the range $[-40^\circ : 40^\circ]$) and encoded using Gaussian probability distributions (with a fixed standard deviation of 10°). These population codes were corrupted with poisson noise. The statistically optimum estimate of C was found using the maximum likelihood estimate of A and B calculated from the noisy input population codes. The network's estimate of C was also calculated by taking the maximum likelihood estimate from the reconstruction generated by the PC/BC-DIM algorithm. Across 100 000 trials, the variance between the true value of C and the network's estimate of C was only 0.01% worse than the variance between the true value of C and the maximum likelihood estimate calculated using the input distributions. Repeating this experiment to estimate B given A and C, found that the network's estimate was 0.08% poorer than the statistically optimum estimate. In comparison, Deneve et al. (2001) report corresponding values for their algorithm of 3.3% and 2.1%.

Further experiments were performed in which the input distributions encoding the values of A and B were given randomly selected standard deviations in addition to randomly selected means (as in the experiments reported in previous sections). The mean was chosen from the range $[-40^\circ : 40^\circ]$ and the standard deviation chosen from the range $[10^\circ : 20^\circ]$. Each input was encoded using a population code corrupted with poisson noise. Fig. 11f plots the network's estimate of the mean of the probability distribution for variable C compared to the statistically optimal estimate for 100 trials. Fig. 11g shows a similar comparison for the variance of the probability distribution encoding variable C. It can be seen that the PC/BC-DIM network produces accurate estimates of both parameters. To quantify these results, the maximum, median, and mean absolute difference between the network's and the optimal estimate of the mean was 0.04° , 0.009° , and 0.01° over 100 000 trials. The maximum, median, and mean percentage absolute difference between the network's and the optimal estimate of the variance was 11.79%, 4.63%, and 4.99% over the same 100 000 trials. Results for an equivalent experiment to estimate B given A and C, are shown in Figs. 11h and i. In this case, the maximum, median, and mean absolute difference



between the network's and the optimal estimate of the mean was 0.52° , 0.03° , and 0.05° over 100 000 trials. The maximum, median, and mean percentage absolute difference between the network's and the optimal estimate of the variance was 13.60%, 5.43%, and 5.67% over the same 100 000 trials. While the poor performance in estimating the variance is a clear limitation of the PC/BC-DIM model, the results presented here still go beyond those reported for previous methods. Specifically, the attractor network model (Deneve et al., 2001; Pouget et al., 2002) is unable to correctly calculate the uncertainty of each variable as the width of each output distribution is fixed (Pouget et al., 2003). Hence, this model can not estimate the variance of the posterior and it would fail on the experiment reported in Fig. 11g as well as the experiment reported in Fig. 11i. Other existing methods for function approximation (Beck et al., 2011; Pouget and Sejnowski, 1997; Pouget and Snyder, 2000) can correctly represent the variance of the posterior, but are limited to performing function approximation in one direction (e.g. calculating C from A and B), and hence, would not be able to perform the experiment shown in Fig. 11b or provide any estimate of either the mean or the variance of B, like that shown in Fig. 11h and i.

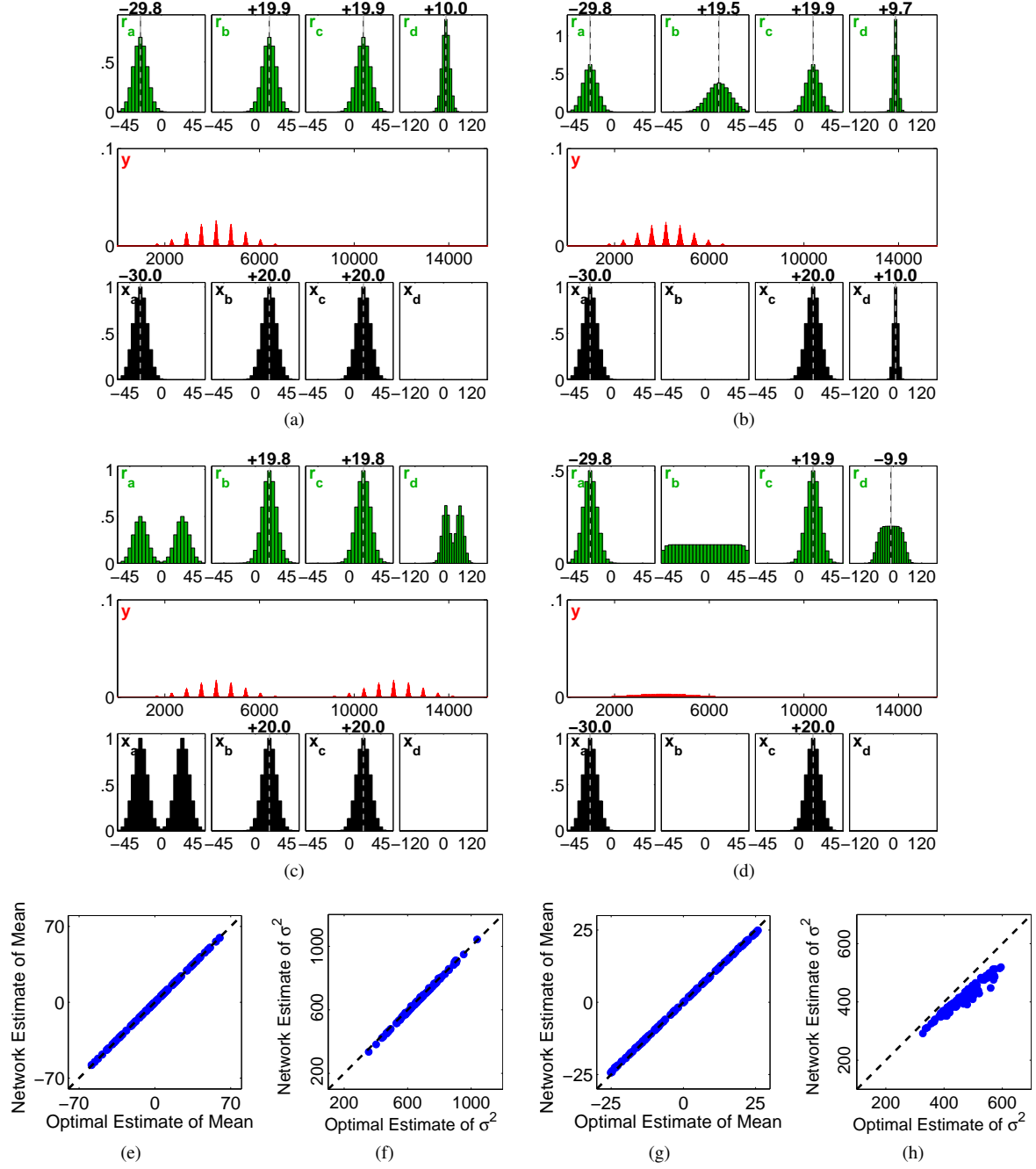
Fig. 12 shows results for a PC/BC-DIM network performing function approximation with four variables (A,

Figure 11: (previous page) Function approximation with three variables. A PC/BC-DIM network as shown in Fig. 1b is used, where the three partitions of the input are used to represent probability distributions for three different variables. If these variables are denoted as A, B, and C, then the network has been wired-up to approximate $C=A+B$. Note that C has a wider range of possible values than A and B, and hence, the x-axes of the histograms representing C have a different scale than those representing A and B. (a) When the two inputs representing A and B are presented (lower histograms), the reconstruction neurons generate an output (upper histograms) that represents the correct estimate of the value of C (as well as outputs representing the given values of A and B). (b) When the two inputs representing A and C are presented (lower histograms), the reconstruction neurons generate an output (upper histograms) that represents the correct estimate of the value of B (as well as outputs representing the given values of A and C). In (a) and (b) the population codes generated by the reconstruction neurons have means which correctly represent the maximum likelihood estimate of the corresponding stimulus value and standard deviations that reflect the certainty in this estimate, such that the estimated values (C in (a) and B in (b)) are represented by population codes with larger variance, although this variance is less than the optimal value. (c) and (d) When all inputs are presented simultaneously to the network, it has multiple (potentially conflicting) estimates of the true value for C. The network performs cue integration with these separate sensory inputs. (c) the inputs to the first two partitions are consistent with a value of C equal to -10 while the input to the third partition indicates that the most likely value of C is +10. In this case, the optimal, combined, estimate of the true value of C is 0. (d) As for (c) except that the precisions of the input cues are no longer equal. The precision of the input to the third partition has been reduced, and hence, the combined estimate of C is now weighted more towards the estimate given by A+B. (e) Gain modulation. Here the response of a single prediction neuron has been measured. Its response is plotted as a function of the value of variable A, for a number of different values of variable B. The position and width of the tuning curve is unaffected by the value of B, but the gain of the response is affected by B. Such gain modulation is observed in various regions along the dorsal pathway, for example, when a retinal RF (variable A) is modulated by eye position (variable B). (f) 100 experiments were performed using input distributions for A and B with randomly chosen means and standard deviations that were encoded with noisy population codes. Each dot shows a comparison between the PC/BC-DIM network's estimate of the mean of the probability distribution encoding the estimated value of C and the optimal estimate for C. (g) As for (f) but for variance. (h) and (i) As for (f) and (g) except using inputs encoding A and C and estimating the probability distribution for B.

B, C, and D). The network encodes the relationship $A+B+C=D$. When any three variables are presented as population codes to the network, it calculates a Gaussian probability distribution encoded by the firing rates of the reconstruction neurons that represents the correct value of the missing variable (Fig. 12a and b). Unlike previous methods of function approximation (Deneve et al., 2001; Pouget et al., 2003, 2002; Pouget and Sejnowski, 1997; Pouget and Snyder, 2000), PC/BC-DIM can simultaneously represent multiple stimuli. It can therefore simultaneously calculate two (or more) separate results, encoded as a bi-modal (or multi-modal) probability distribution, as illustrated in Fig. 12c.

When fewer than three inputs are present the uncertainty about the missing values is large, and this is reflected in the population codes calculated by the network (as shown in Fig. 12d). However, as for the previous network (with three inputs) the variance of the output distributions is underestimated. Specifically, the probability distribution for B should be uniform over the full range of possible values $[-60^\circ : 60^\circ]$. However, edge effects result in the probability distribution not being completely uniform. Given that B has an equal probability of taking any value in the range -60° to $+60^\circ$, and zero probability of taking a value from outside this range, the posterior for D should be uniform in the range -70° to $+50^\circ$. The population code produced by the reconstruction neurons is approximately uniform between -50° and $+30^\circ$ which is less than the expected range (and is due to the underestimation of the range of B). However, as noted earlier in this section, previous algorithms that perform function approximation (Beck et al., 2011; Deneve et al., 2001; Pouget et al., 2002; Pouget and Sejnowski, 1997; Pouget and Snyder, 2000) would completely fail on this task.

Figs. 12e and f compare the network's estimate of the mean and variance of the probability distribution encoding variable D with the statistically optimal estimates of these parameters. These results are for 100 trials with the values for A, B and C chosen at random (uniformly in the range $[-25^\circ : 25^\circ]$) and encoded using Gaussian probability distributions (with a standard deviation chosen at random from the range $[10^\circ : 20^\circ]$). The equivalent results for B, estimated from inputs encoding A, C, and D are shown in Figs. 12g and h. It can be seen that PC/BC-DIM is accurate at calculating the parameters of each posterior distribution except that it underestimates the variance of the probability distribution encoding variable B.



It is possible to wire-up PC/BC-DIM networks to encode any function defined over any number of variables. However, the number of prediction neurons required increases exponentially with the number of variables, as is the case for any other method that computes with basis functions (Deneve and Pouget, 2003; Pouget and Sejnowski, 1997). To resolve this issue it is theoretically possible to decompose computations into several steps, and implement each sub-task using a separate basis function network (Pouget et al., 2002). While previous methods of computing with basis functions (e.g., Deneve et al., 2001; Deneve and Pouget, 2003; Pouget et al., 2002; Pouget and Sejnowski, 1997; Pouget and Snyder, 2000) should be capable of operating in this way, it has not been demonstrated that they can. Here, it is shown that a PC/BC-DIM network can be decomposed into multiple sub-networks to compute a function. Fig. 13a shows a single PC/BC-DIM network performing function approximation with four variables (A, B, C, and D). While it is possible to provide inputs to any of the four partitions, and read outputs from any of the four partitions of the reconstruction neurons, the particular combination of inputs and outputs needed to estimate D given A, B and C is shown in Fig. 13a. The architecture required to allow the same function to be approximated with two interconnected PC/BC-DIM networks, forming a simple two-stage PC/BC-DIM

Figure 12: (previous page) Function approximation with four variables. A PC/BC-DIM network has been wired-up to approximate the function $D=A+B+C$. Note that D has a wider range of possible values than A , B , and C , and hence, the x-axes of the histograms representing D have different scales to those representing A , B , and C . (a) When the three inputs representing A , B , and C are presented (lower histograms), the reconstruction neurons generate an output (upper histograms) that represents the correct value of D (as well as outputs representing the given values of A , B , and C). (b) When the three inputs representing A , C , and D are presented (lower histograms), the reconstruction neurons generate an output (upper histograms) that estimates the correct value of B (as well as outputs representing the given values of A , C , and D). (c) As (a) but with two values of A represented by a bi-modal population code presented to the first partition of the input. The network correctly calculates two values for D represented by the bi-modal population code produced by the reconstruction neurons in the last partition. (d) When the two inputs representing A and C are presented (lower histograms), there is a large uncertainty about the values of B and D and this is (partially) represented in the probability distributions encoding these variables generated by the network (upper histograms). (e) 100 experiments were performed using input distributions for A , B , and C with randomly chosen means and standard deviations, that were encoded with noisy population codes. Each dot shows a comparison between the PC/BC-DIM network's estimate of the mean of the probability distribution encoding the approximated value of D and the optimal estimate for D . (f) As for (e) but for variance. (g) and (h) As for (e) and (f) except using inputs encoding A , C , and D and estimating the parameters of the probability distribution for variable B .

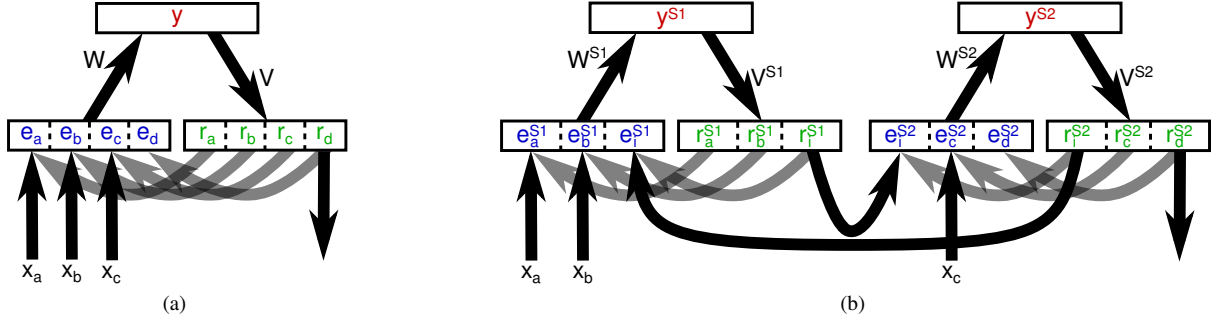


Figure 13: PC/BC-DIM neural network architectures for function approximation with four variables. (a) A single-stage network to calculate D given A , B , and C . (b) A hierarchical architecture, consisting of two interconnected PC/BC-DIM networks, for calculating the same function.

hierarchy, is shown in Fig. 13b. The first network calculates an intermediate result ($A+B$) in one partition of its reconstruction neurons. This intermediate result provides input to one of the partitions of the second PC/BC-DIM network. The second network's reconstruction of this intermediate representation is fed-back as input to the first PC/BC-DIM network. The resulting two-stage network has fewer prediction neurons in total than the equivalent single-stage network (1850 compared to 15625 for the particular task used here), however, it produces almost identical results (not shown) to those presented in Fig. 12.

To quantitatively compare the performance of the single-stage network and the hierarchical network, both networks were used to perform function approximation with noisy input population codes. 100 000 trials were performed for two conditions. In the first condition each network calculated D given noisy inputs encoding randomly selected values for A , B , and C . The statistically optimum estimate of D was found using the maximum likelihood estimate of A , B , and C calculated from the noisy input population codes and this was compared to the network's estimate of D calculated by taking the maximum likelihood estimate from the reconstruction generated by the PC/BC-DIM algorithm. The median absolute difference between the network's estimate and the optimal estimate was 0.14° for the single-stage network, and 0.005° for the hierarchical network. The median percentage absolute difference between the network's estimate of the variance of the probability distribution encoding D and the statistically optimum estimate was 1.48% for the single-stage network, and 5.02% for the hierarchical network. In the second condition, each network calculated B given noisy inputs encoding randomly selected values for A , C , and D . Across 100 000 trials, the median absolute difference between the network's estimate and the optimal estimate of the mean was 0.38° for the single-stage network, and 0.29° for the hierarchical network. The median percentage absolute difference between the network's estimate and the statistically optimum estimate of the variance was 9.75% for the single-stage network, and 2.83% for the hierarchical network. Hence, both the

single-stage and hierarchical networks produce reasonable estimates of the posterior distributions.

3.6 Computations with Non-Gaussian Distributions

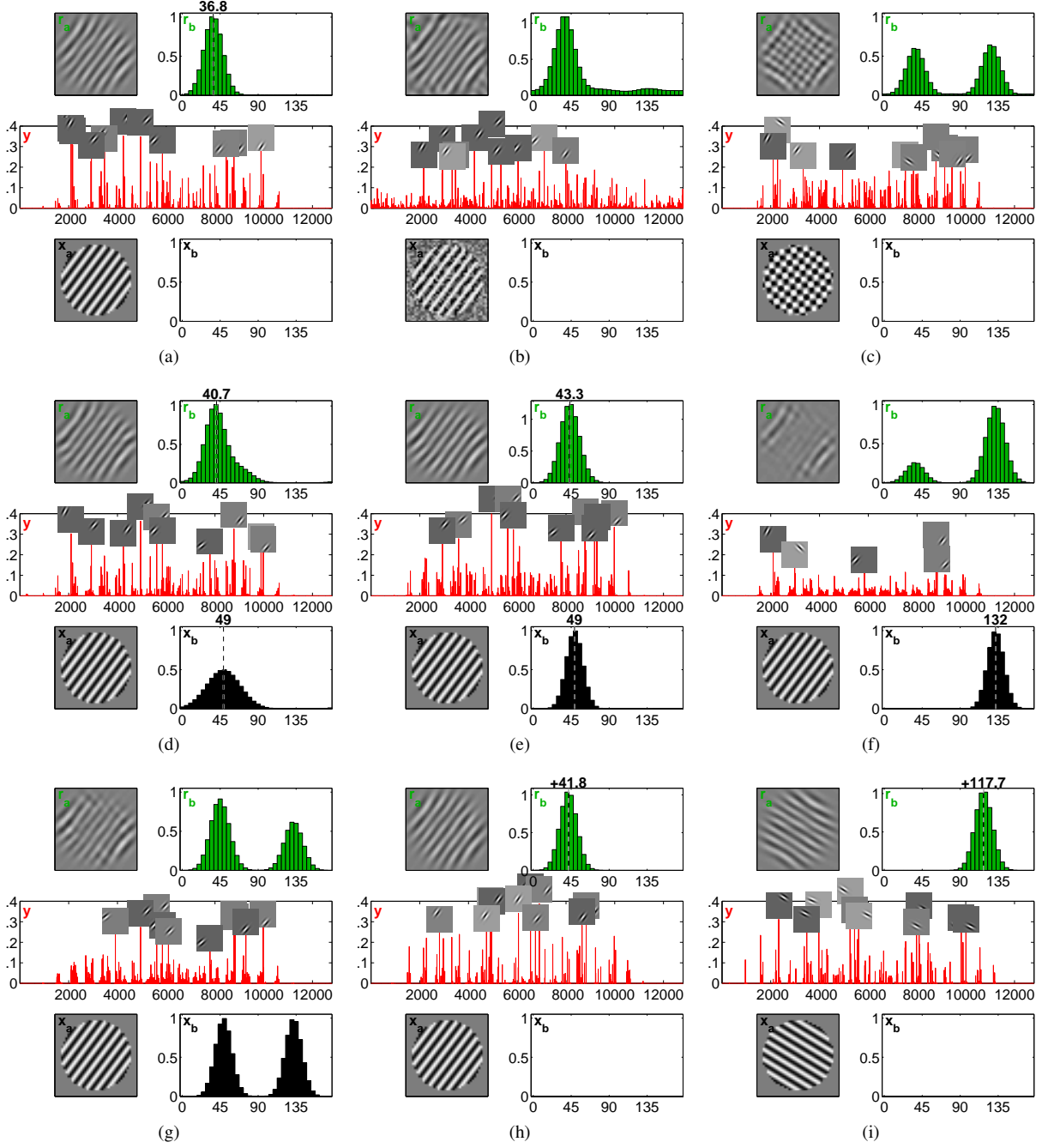
All the previous experiments have been performed using one-dimensional Gaussian population codes, and networks with Gaussian RFs. The results in this section demonstrate that PC/BC-DIM can perform Bayesian inference with stimuli and synaptic weights that are not Gaussian. The input is a greyscale image and the prediction neurons are given synaptic weights defined using Gabor functions^c. The RFs of the whole population of prediction neurons tile the input image with Gabor RFs. A second set of inputs (a one-dimensional vector) is defined to represent orientation. Each prediction neuron receives a Gaussian RFs from this second partition of the input, with this Gaussian centred at the orientation corresponding to the orientation of that prediction neuron's Gabor RF^d. Defining this extra partition of the input automatically defines a second partition of the reconstruction neuron population. The reconstruction in this second partition will be a population code representing the distribution of orientations signaled by the responses of the prediction neurons. It would be possible to define further partitions of the input and reconstruction neuron population to encode other features represented by the prediction neurons (*e.g.*, location or phase), however, here we only consider stimulus orientation. The reconstruction neurons that represent orientation are like complex-cells in V1 as they pool the responses of multiple prediction neurons (corresponding to simple-cells in this analogy). The orientation-selective reconstruction neurons pool the responses of prediction neurons with the same orientation preference, but with RFs at a range of spatial locations and with a range of phase preferences. Here, so that we only have one population code representing orientation, spatial pooling takes place over the whole image rather than in a small patch of image as would be the case for cortical complex-cells.

If an input image is presented to this PC/BC-DIM network it generates reconstruction neuron responses that represent both the image and a population code representing a probability distribution of orientations within the image, as illustrated in Fig. 14. When the image contains a single sinusoidal grating, the population code is a Gaussian with a peak approximately at the orientation of the grating (Fig. 14a). If the image is corrupted by noise, then the population code is more distributed (Fig. 14b). If a second sinusoidal grating is superimposed over the first, then the population code is bi-modal with peaks at approximately the orientation of both gratings (Fig. 14c). If both an input image and a population code representing orientation are simultaneously presented as inputs of the network, then cue integration can occur. For example, if the input image is a single sinusoidal grating oriented at 37° from the vertical, and the population code is a Gaussian distribution centred at 49° , then the combined estimate of the orientation represented by the second partition of the reconstruction neurons is intermediate between the values signalled by the two cues (Fig. 14d). Furthermore, increasing the precision of the input probability distribution (by reducing its standard deviation from 20° to 10°) causes the combined estimate to shift further towards 49° (Fig. 14e). When cue conflict is large, the network performs cue segregation rather than integration, generating a bi-modal distribution of responses in the second partition of reconstruction neurons. This distribution has peaks at approximately the orientation of both cues (Fig. 14f). If multiple cues are supplied then both cue integration and segregation can co-occur (Fig. 14g). To encode a prior into the weights of the network, the Gaussian weights for each prediction neuron were scaled by a function of orientation (the prior). This function of orientation was a Gaussian centred at 90° and with a standard deviation of 30° . When the image contains a single sinusoidal grating, the population code representing orientation that is generated by the network is shifted towards 90° (Fig. 14h and i).

The above results are qualitatively consistent with Bayesian inference. To determine if the inference performed by the network is optimal it would be necessary to compare the posterior calculated by the network with that expected from exact Bayesian inference. However, the form of the true likelihood for the image cue, and hence, the correct posterior is not known. The situation is analogous to that faced when psychophysical experiments are performed to assess if human performance on cue integration tasks is optimal (*e.g.*, Battaglia et al., 2003; Ernst and Banks, 2002; Helbig and Ernst, 2007; Jacobs, 1999; Knill and Saunders, 2003). In such experiments it is assumed that the probability distributions take a certain form (typically that they are all Gaussian), and the parameters of the distributions encoding each cue (*i.e.*, the means and variances) are then estimated from experiments in which those cues are presented to the subject in isolation. From these estimates the maximum-likelihood estimate for the cue integration task is determined and this value is compared to the subject's response when presented with both cues. The same procedure can be followed for the current simulation results. It is assumed that the image cue

^cAs in previous work using PC/BC-DIM to model V1 (Spratling, 2010, 2011, 2012a,b,c), the input is divided into ON and OFF channels containing the high and low contrast components of the image, respectively, and the positive and negative values of the Gabor function are used to define separate RFs for these two input channels. The figures show the original image, which is equal to the ON channel minus the OFF channel of the input to PC/BC-DIM, and the reconstruction of the ON channel minus the reconstruction of the OFF channel.

^dIn contrast to previous experiments, here the \mathbf{V} weights for each partition were scaled independently to allow both cues to have a similar influence on prediction neuron response.



is equivalent to a Gaussian probability distribution with a mean given by the orientation of the image and with a standard deviation of 11.2° . The latter value is an estimate derived from the width of the posterior produced by the reconstruction neurons when only the image cue is presented to the PC/BC-DIM network (as in Fig. 14a). 100 cue integration trials were performed with randomly selected cues. In each trial the first cue was an image of a single sinusoidal grating at a randomly chosen orientation, and the second cue was a Gaussian population code representing an orientation with a randomly chosen cue conflict of up to $\pm 10^\circ$, and a random standard deviation, chosen uniformly from the range $[10^\circ : 20^\circ]$. In each case the optimal estimate of the orientation was calculated assuming the first cue was equivalent to a Gaussian probability distribution centred at the orientation of the grating and with a standard deviation of 11.2° . When this value was compared to the network's estimate of the orientation, they were found to be in very close agreement (Fig. 15a). Over the 100 trials the maximum, median, and mean absolute difference between the network's and the optimal estimate of the orientation was 0.63° , 0.20° , and 0.22° .

The preceding experiment was repeated using input images corrupted with noise, like that shown in Fig. 14b, as the first cue and Gaussian input distributions corrupted by poisson noise as the second cue. In this case, to

Figure 14: (previous page) Decoding, cue integration, cue segregation, and Bayesian inference with a prior using image stimuli and neurons with Gabor RFs. The format of the diagrams is the same as used in previous figures, except here the inputs to, and the reconstructions of, the first partition are shown as 2D images rather than 1D vectors. Also, here, the RFs of the most active prediction neurons are indicated by the grey squares superimposed on the middle histograms. The numbers above the histograms show the maximum likelihood estimate of the stimulus value calculated from that histogram (equation 6). (a) The input image (lower left) is a circular sinusoidal grating oriented at 37° from the vertical. The reconstruction neurons generate a population code (upper right histogram) that represents the orientation of this stimulus. (b) as (a) but the input image is corrupted with speckle noise. The reconstruction neurons' representation of the orientation (upper right histogram) is peaked at approximately the correct position, but is wider than in (a). Compare this with the decoding of noisy Gaussian population codes illustrated in Fig. 2. (c) The input image (lower left) is composed of two superimposed sinusoidal gratings, one oriented at 37° and the other at 127° from the vertical. The reconstruction neurons' representation of the orientation (upper right histogram) is bi-modal and correctly represents the orientations of both gratings (*cf.*, Fig. 4). (d) and (e) The input image is the same as in (a) but a second Gaussian population code, representing orientation, is also presented to the second partition of the input (lower right). The PC/BC-DIM network integrates both cues to generate a combined estimate of the orientation (upper right histogram). When the Gaussian population code that is provided as input to the second partition has greater precision, as shown in (e), the estimate of the orientation moves towards the value indicated by that cue. Compare these results with cue integration for Gaussian population codes illustrated in Fig. 6. (f) When the cue conflict between the orientation of the image and that encoded by the Gaussian population code is large, cue segregation occurs and the reconstruction neurons' representation of the orientation (upper right histogram) is bi-modal and correctly represents the orientations of both cues (*cf.*, Fig. 9a). (g) The orientation input contains two cues (those used in both (e) and (f)) represented by a bi-modal population code. The first orientation cue is integrated with the orientation information extracted from the image, the second orientation cue is represented by a separate peak in the reconstruction neurons' representation of the orientation (upper right histogram). Compare this with simultaneous cue integration and cue segregation for Gaussian population codes illustrated in Fig. 9b. (h) and (i) To incorporate a prior the amplitude of each prediction neuron's Gaussian RF has been modulated by a Gaussian centred at an orientation of 90° from the vertical and with a standard deviation of 30° . The input (lower left images) is a circular sinusoidal grating oriented at 37° (h) and 121.5° (i) from the vertical. Due to the prior being centred at 90° the network's estimates of the posterior probability distributions (upper right histograms) are shifted towards 90° . Compare this with Bayesian inference using Gaussian population codes illustrated in Fig. 5.

calculate the optimal estimate of the orientation it was assumed that the image was equivalent to a Gaussian probability distribution with a standard deviation of 13.6° . Again, these optimal estimates were found to be consistent with the network's estimates of the orientation in this cue combination task, as shown in Fig. 15b. Over the 100 trials the maximum, median, and mean absolute difference between the network's and the optimal estimate of the orientation was 1.40° , 0.40° , and 0.43° . The accuracy of the estimates of the variance of the posterior were comparable to the same experiment performed with two cues defined by Gaussian probability distributions (see Section 3.3). Specifically, over the 100 trials the maximum, median, and mean percentage absolute difference between the network's and the optimal estimate of the variance was 19.78%, 4.13%, and 5.78%.

Performing 100 trials with the only input being an image of a single sinusoidal grating at a randomly chosen orientation, but with a prior imposed on the weights (like in the simulation results shown in Fig. 14h and i), also produced estimates of the orientation close to the statistically optimum value that would be predicted by applying Bayes theorem (as shown in Fig 15c). Specifically, the maximum, median, and mean absolute difference between the network's and the optimal estimate of the orientation was 1.59° , 0.53° , and 0.64° . Hence, despite one cue being defined in terms of a two-dimensional array of intensity values, rather than a one-dimensional Gaussian population code, the PC/BC-DIM network is still capable of performing accurate inference.

4 Discussion

Recently Bayesian theories of cognition have been heavily criticised (*e.g.*, Bowers and Davis, 2012; Jones and Love, 2011; Marcus and Davis, 2013). The PC/BC-DIM model addresses many of these criticisms. For example, one criticism is that human behaviour is seldom rational and optimal, and hence, there is little evidence that the brain performs optimal, Bayesian, inference (Bowers and Davis, 2012; Jones and Love, 2011). However, PC/BC-

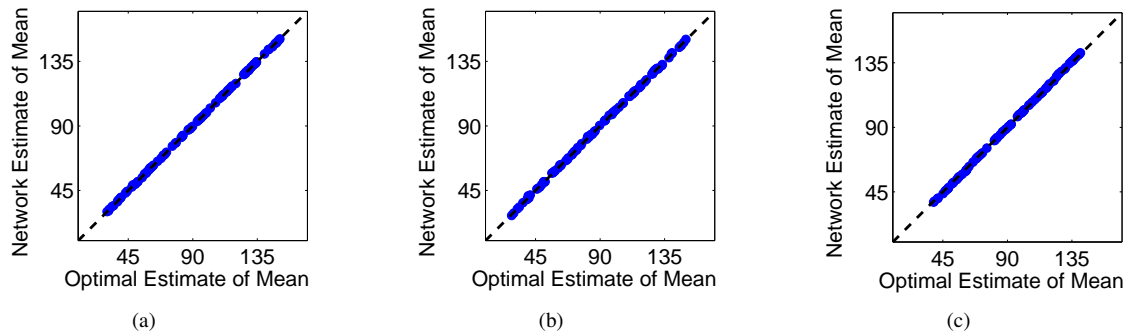


Figure 15: Accuracy of inference when using image stimuli and neurons with Gabor RFs. (a) and (b) Cue integration accuracy. Each dot shows a comparison of the PC/BC-DIM network’s estimate of the mean of the combined probability distribution compared to the probabilistically optimal estimate of the mean calculated by assuming that the probability distribution defined by the image cue is equivalent to a Gaussian. 100 experiments were performed with randomly chosen image orientations, cue conflicts and precisions for the Gaussian cue. In (a) there is no noise in the inputs, in (b) the image cue and the Gaussian cue are corrupted by noise. (c) Accuracy of the posterior calculated from a likelihood and a prior. The prediction neuron RFs were modified to incorporate a Gaussian prior centred at an orientation of 90° and with a standard deviation of 30° . Each dot shows a comparison of the PC/BC-DIM network’s estimate of the posterior compared to the probabilistically optimal estimate obtained by assuming that the image is equivalent to a Gaussian probability density with standard deviation 11.2° . 100 experiments were performed with randomly chosen image orientations.

DIM proposes that the brain is engaged in predictive coding (Clark, 2013; Huang and Rao, 2011; Rao and Ballard, 1999), and that Bayesian inference is just one of the functions that can be achieved by predictive coding. Exact Bayesian inference may be implemented in the brain, using predictive coding, only in specific circumstances, allowing people to act optimally when their mental models of the environment are veridical, or to reason and act optimally with respect to the sub-optimal models that they possess (Jones and Love, 2011). Jones and Love (2011) point out that “the most substantial part of learning lies in constructing a generative model of one’s environment.” Building such models is a task that predictive coding is particularly suited for (Clark, 2013).

Another criticism is that Bayesian models are tested using a breadth-first strategy that provides a superficial explanation of a range of carefully selected tasks, rather than using a depth-first strategy where a model is tested in detail in a challenging domain (Marcus and Davis, 2013). In contrast, PC/BC-DIM has been tested, in-depth, as a model of V1 and has been shown to provide a comprehensive account of primary visual cortex function (Spratling, 2010, 2011, 2012a,c). A further criticism is that different models of Bayesian inference have been used to simulate different tasks (Marcus and Davis, 2013). In contrast, PC/BC-DIM has been used to simulate a range of probabilistic inference tasks in the current work and a wide range of other neurophysiological and cognitive processes in previous work (as listed in the Introduction). An additional criticism of Bayesian models is that they have so many free parameters (such as the choice of priors, generative model, *etc.*) that they can simulate any behaviour, and that these parameters are altered, post hoc, to fit the data (Bowers and Davis, 2012; Jones and Love, 2011; Marcus and Davis, 2013). PC/BC-DIM also has many free parameters, principally the synaptic weights of the prediction neurons. However, in the PC/BC-DIM model of V1 these parameters were either learnt from natural images, or were defined to be Gabor-like, and hence, to resemble the RFs of V1 neurons. The parameters were, therefore, not chosen arbitrarily to fit the data, and were kept fixed across numerous experiments.

A further criticism of Bayesian brain models is that they are defined at the computational level of analysis and make no predictions about, nor are constrained by, neural mechanisms and psychological processes (Bowers and Davis, 2012; Jones and Love, 2011). However, previous work has described, biologically-plausible, neurally-based implementations of Bayesian inference. For example, several models have proposed how priors can be combined with likelihoods to calculate posteriors (Ganguli and Simoncelli, 2010, 2014; Girshick et al., 2011; Shi and Griffiths, 2009), however, these models fail to perform other probabilistic computations such as cue integration or function approximation. Other models do perform cue integration (Ma et al., 2006) but fail to perform function approximation and vice versa (Beck et al., 2011). Still other models can perform both cue integration and function approximation (Deneve et al., 1999, 2001; Latham et al., 2003; Pouget et al., 2003, 2002, 2000, 1998), but fail to calculate the variance of the posterior, are restricted to working with mono-modal Gaussian distributions, and can not perform Bayesian inference with a non-uniform prior. This article proposes an alternative

neural implementation of Bayesian inference that overcomes the limitations of these previous methods. The PC/BC-DIM model is particularly simple while providing a particularly comprehensive account of probabilistic computation, that includes: inference with priors; inference with noisy population codes; hierarchical inference; inference with more than one stimulus or cause; inference with non-Gaussian stimuli and non-Gaussian RFs; cue integration; cue segregation; and function approximation. However, there remain a number of limitation of the PC/BC-DIM model of probabilistic inference. Firstly, in all tasks the response of the reconstruction neurons represents the posterior except in cue integration tasks where the posterior has been encoded by the response of the reconstruction neurons raised to a power equal to the number of cues. Secondly, the estimate of the variance of the posterior can be inaccurate, particularly when the PC/BC-DIM algorithm is used to perform certain function approximations. Thirdly, the PC/BC-DIM model also fails to account for the capacity of the cortex to make fine distinctions between stimuli using neurons with broadly tuned RFs. Finally, the current paper fails to provide formal, mathematical, insights into why PC/BC-DIM succeeds in performing Bayesian inference.

Acknowledgements

Thanks to the organisers of, and the participants at, the Lorentz Centre Workshop on Perspectives on Human Probabilistic Inference (May 2014) for discussions that inspired this work. Thanks also to Kris De Meyer and the anonymous reviewers for helpful comments on earlier drafts of this paper.

References

- Achler, T. (2014). Symbolic neural networks for cognitive capacities. *Biologically Inspired Cognitive Architectures*, 9(0):71–81.
- Achler, T. and Amir, E. (2008). Input feedback networks: Classification and inference based on network structure. In Wang, P., Goertzel, B., and Franklin, S., editors, *Artificial General Intelligence*, pages 15–26, Amsterdam, The Netherlands. IOS Press.
- Alger, B. E. (2002). Retrograde signaling in the regulation of synaptic transmission: focus on endocannabinoids. *Progress in Neurobiology*, 68(4):247–86.
- Alink, A., Schwiedrzik, C. M., Kohler, A., Singer, W., and Muckli, L. (2010). Stimulus predictability reduces responses in primary visual cortex. *The Journal of Neuroscience*, 30:2960–6.
- Anastasio, T. J., Patton, P. E., and Belkacem-Boussaid, K. (2000). Using Bayes’ rule to model multisensory enhancement in the superior colliculus. *Neural Computation*, 12:1165–87.
- Anderson, C. H. and Van Essen, D. C. (1994). Neurobiological computational systems. In *IEEE World Congress on Computational Intelligence*, pages 213–22.
- Ballard, D. H. and Jehee, J. (2012). Dynamic coding of signed quantities in cortical feedback circuits. *Frontiers in Psychology*, 3:254.
- Barbas, H. and Rempel-Clower, N. (1997). Cortical structure predicts the pattern of corticocortical connections. *Cerebral Cortex*, 7:635–46.
- Barber, M. J., Clark, J. W., and Anderson, C. H. (2003). Neural representation of probabilistic information. *Neural Computation*, 15(8):1843–64.
- Barlow, H. B. (1969). Pattern recognition and the responses of sensory neurons. *Annals of the New York Academy of Sciences*, 156:872–81.
- Barone, P., Batardiere, A., Knoblauch, K., and Kennedy, H. (2000). Laminar distribution of neurons in extrastriate areas projecting to visual areas V1 and V4 correlates with the hierarchical rank and indicates the operation of a distance rule. *The Journal of Neuroscience*, 20:3263–81.
- Battaglia, P. W., Jacobs, R. A., and Aslin, R. N. (2003). Bayesian integration of visual and auditory signals for spatial localization. *Journal of the Optical Society of America. A, Optics, Image Science, and Vision*, 20(7).
- Beck, J., Heller, K., and Pouget, A. (2012). Complex inference in neural circuits with probabilistic population codes and topic models. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 25, pages 3059–67. Curran Associates, Inc.
- Beck, J. M., Latham, P. E., and Pouget, A. (2011). Marginalization in neural circuits with divisive normalization. *The Journal of Neuroscience*, 31(43):15310–9.
- Beierholm, U. R., Kording, K. P., Shams, L., and Ma, W. J. (2008). Comparing Bayesian models for multisensory cue combination without mandatory integration. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems*, volume 20, pages 81–8. Curran Associates, Inc.
- Bowers, J. S. and Davis, C. J. (2012). Bayesian just-so stories in psychology and neuroscience. *Psychological Bulletin*, 138(3):389–414.

- Branco, T. and Staras, K. (2009). The probability of neurotransmitter release: variability and feedback control at single synapses. *Nature Reviews Neuroscience*, 10:373–83.
- Brozović, M., Abbott, L. F., and Andersen, R. A. (2008). Mechanism of gain modulation at single neuron and network levels. *Journal of Computational Neuroscience*, 25:158–68.
- Budd, J. M. L. (1998). Extrastriate feedback to primary visual cortex in primates: a quantitative analysis of connectivity. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 265(1400):1037–44.
- Carandini, M. and Heeger, D. J. (1994). Summation and division by neurons in primate visual cortex. *Science*, 264(5163):1333–6.
- Chance, F. S. and Abbott, L. F. (2000). Divisive inhibition in recurrent networks. *Network: Computation in Neural Systems*, 11:119–29.
- Chater, N., Tenenbaum, J. B., and Yuille, A. (2006). Probabilistic models of cognition: conceptual foundations. *Trends in Cognitive Sciences*, 10:287–91.
- Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(03):181–204.
- Crick, F. and Koch, C. (1998). Constraints on cortical and thalamic projections: the no-strong-loops hypothesis. *Nature*, 391:245–50.
- De Meyer, K. and Spratling, M. W. (2011). Multiplicative gain modulation arises through unsupervised learning in a predictive coding model of cortical function. *Neural Computation*, 23(6):1536–67.
- De Meyer, K. and Spratling, M. W. (2013). A model of partial reference frame transforms through pooling of gain-modulated responses. *Cerebral Cortex*, 23(5):1230–9.
- Deneve, S. (2008). Bayesian spiking neurons I: Inference. *Neural Computation*, 20(1):91–117.
- Deneve, S., Latham, P. E., and Pouget, A. (1999). Reading population codes: a neural implementation of ideal observers. *Nature Neuroscience*, 2(8):740–5.
- Deneve, S., Latham, P. E., and Pouget, A. (2001). Efficient computation and cue integration with noisy population codes. *Nature Neuroscience*, 4(8):826–31.
- Deneve, S. and Pouget, A. (2003). Basis functions for object-centered representations. *Neuron*, 37:347–59.
- Egner, T., Monti, J. M., and Summerfield, C. (2010). Expectation and surprise determine neural population responses in the ventral visual stream. *The Journal of Neuroscience*, 30(49):16601–8.
- Ernst, M. and Jäkel, F. (2003). Learning to combine arbitrary signals from vision and touch. In *4th International Multisensory Research Forum*.
- Ernst, M. O. and Banks, M. S. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415:429–33.
- Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in primate cerebral cortex. *Cerebral Cortex*, 1:1–47.
- Fiser, J., Berkes, P., Orban, G., and Lengyel, M. (2010). Statistically optimal perception and learning: from behavior to neural representations. *Trends in Cognitive Sciences*, 14(3):119–30.
- Földiák, P. (1993). The 'ideal homunculus': statistical inference from neural population responses. In Eeckman, F. and Bower, J., editors, *Computation and Neural Systems: Proceedings of the Computational Neuroscience Meeting*, pages 55–60, London, UK. Kluwer Academic Publishers.
- Friston, K. J. (2005). A theory of cortical responses. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 360(1456):815–36.
- Gabbiani, F., Krapp, H. G., Koch, C., and Laurent, G. (2002). Multiplicative computation in a visual neuron sensitive to looming. *Nature*, 420:320–4.
- Ganguli, D. and Simoncelli, E. P. (2010). Implicit encoding of prior probabilities in optimal neural populations. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 23, pages 658–66. Curran Associates, Inc.
- Ganguli, D. and Simoncelli, E. P. (2014). Efficient sensory encoding and bayesian inference with heterogeneous neural populations. *Neural Computation*, 26(10):2103–34.
- Georgopoulos, A. P., Schwartz, A. B., and Kettner, R. E. (1986). Neuronal population coding of movement direction. *Science*, 233:1416–9.
- Girshick, A., Landy, M., and Simoncelli, E. (2011). Cardinal rules: Visual orientation perception reflects knowledge of environmental statistics. *Nature Neuroscience*, 14(7):926–32.
- Griffiths, T. L., Kemp, C., and Tenenbaum, J. B. (2008). Bayesian models of cognition. In Sun, R., editor, *Cambridge Handbook of Computational Cognitive Modeling*. Cambridge University Press, Cambridge, UK.
- Griffiths, T. L. and Tenenbaum, J. B. (2006). Optimal predictions in everyday cognition. *Psychological Science*, 17(9):767–73.
- Harpur, G. F. (1997). *Low Entropy Coding with Unsupervised Neural Networks*. PhD thesis, Department of Engineering, University of Cambridge.

- Heeger, D. J. (1992). Normalization of cell responses in cat striate cortex. *Visual Neuroscience*, 9:181–97.
- Helbig, H. B. and Ernst, M. O. (2007). Optimal integration of shape information from vision and touch. *Experimental Brain Research*, 179(4):595–606.
- Huang, Y. and Rao, R. P. N. (2011). Predictive coding. *WIREs Cognitive Science*, 2:580–93.
- Jacobs, R. A. (1999). Optimal integration of texture and motion cues to depth. *Vision Research*, 39:3621–9.
- Jaffe, D. B. and Carnevale, N. T. (1999). Passive normalization of synaptic integration influenced by dendritic architecture. *Journal of Neurophysiology*, 82:3268–85.
- Jazayeri, M. and Movshon, J. A. (2006). Optimal representation of sensory information by neural populations. *Nature Neuroscience*, 9:690–6.
- Johnson, R. R. and Burkhalter, A. (1997). A polysynaptic feedback circuit in rat visual cortex. *The Journal of Neuroscience*, 17(18):7129–40.
- Jones, M. and Love, B. C. (2011). Bayesian fundamentalism or enlightenment? on the explanatory status and theoretical contributions of Bayesian models of cognition. *Behavioral and Brain Sciences*, 34(4):169–188.
- Kersten, D., Mamassian, P., and Yuille, A. (2004). Object perception as Bayesian inference. *Annual Review of Psychology*, 55(1):271–304.
- Knill, D. C. and Richards, W. (1996). *Perception as Bayesian Inference*. Cambridge University Press, Cambridge, UK.
- Knill, D. C. and Saunders, J. A. (2003). Do humans optimally integrate stereo and texture information for judgments of surface slant? *Vision Research*, 43:2539–58.
- Koch, C. and Segev, I. (2000). The role of single neurons in information processing. *Nature Neuroscience*, 3(supplement):1171–7.
- Kok, P. and de Lange, P. F. (2015). Predictive coding in sensory cortex. In Forstmann, U. B. and Wagenmakers, E.-J., editors, *An Introduction to Model-Based Cognitive Neuroscience*, pages 221–44. Springer, New York, NY.
- Kok, P., Rahnev, D., Jehee, J. F. M., Lau, H. C., and de Lange, F. P. (2012). Attention reverses the effect of prediction in silencing sensory signals. *Cerebral Cortex*, 22:2197–206.
- Larkum, M. E., Senn, W., and Lüscher, H.-R. (2004). Top-down dendritic input increases the gain of layer 5 pyramidal neurons. *Cerebral Cortex*, 14(10):1059–70.
- Latham, P. E., Deneve, S., and Pouget, A. (2003). Optimal computation with attractor networks. *Journal of Physiology – Paris*, 97(4–6):683–94.
- Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13, Cambridge, MA. MIT Press.
- Lee, T. S. and Mumford, D. (2003). Hierarchical Bayesian inference in the visual cortex. *Journal of the Optical Society of America. A, Optics, Image Science, and Vision*, 20:1434–48.
- Lochmann, T. and Deneve, S. (2011). Neural processing as causal inference. *Current Opinion in Neurobiology*, 21(5):774–81.
- Lochmann, T., Ernst, U. A., and Denève, S. (2012). Perceptual inference predicts contextual modulations of sensory responses. *The Journal of Neuroscience*, 32(12):4179–95.
- London, M. and Häusser, M. (2005). Dendritic computation. *Annual Review of Neuroscience*, 28:503–32.
- Ma, W. J. (2012). Organising probabilistic models of perception. *Trends in Cognitive Sciences*, 16(10):511–8.
- Ma, W. J., Beck, J., Latham, P. E., and Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nature Neuroscience*, 9(11):1432–8.
- Ma, W. J., Beck, J. M., and Pouget, A. (2008). Spiking networks for Bayesian inference and choice. *Current Opinion in Neurobiology*, 18(2):217–22.
- Ma, W. J. and Jazayeri, M. (2014). Neural coding of uncertainty and probability. *Annual Review of Neuroscience*, 37:205–20.
- Ma, W. J. and Pouget, A. (2008). Linking neurons to behavior in multisensory perception: A computational review. *Brain Research*, 1242(0):4–12.
- Marcus, G. F. and Davis, E. (2013). How robust are probabilistic models of higher-level cognition? *Psychological Science*, 24(12):2351–60.
- Markov, N. T., Vezoli, J., Chameau, P., Falchier, A., Quilodran, R., Huissoud, C., Lamy, C., Misery, P., Giroud, P., Ullman, S., Barone, P., Dehay, C., Knoblauch, K., and Kennedy, H. (2014). Anatomy of hierarchy: Feedforward and feedback pathways in macaque visual cortex. *Journal of Comparative Neurology*, 522(1):225–59.
- Mehaffey, W. H., Doiron, B., Maler, L., and Turner, R. W. (2005). Deterministic multiplicative gain control with active dendrites. *The Journal of Neuroscience*, 25:9968–77.
- Mel, B. W. (1994). Information processing in dendritic trees. *Neural Computation*, 6:1031–85.
- Mitchell, S. J. and Silver, R. A. (2003). Shunting inhibition modulates neuronal gain during synaptic excitation.

- Neuron*, 38(3):433–45.
- Mountcastle, V. B. (1998). *Perceptual Neuroscience: The Cerebral Cortex*. Harvard University Press, Cambridge, MA.
- Murphy, B. K. and Miller, K. D. (2003). Multiplicative gain changes are induced by excitation or inhibition alone. *The Journal of Neuroscience*, 23:10040–51.
- Olsen, S. R., Bortone, D. S., Adesnik, H., and Scanziani, M. (2012). Gain control by layer six in cortical circuits of vision. *Nature*, 483:47–52.
- Phillips, W. A. (2016). On the cognitive functions of intracellular mechanisms for contextual amplification. *Brain and Cognition*, (in press).
- Pouget, A., Beck, J. M., Ma, W. J., and Latham, P. E. (2013). Probabilistic brains: knowns and unknowns. *Nature Neuroscience*, 16:1170–8.
- Pouget, A., Dayan, P., and Zemel, R. S. (2003). Inference and computation with population codes. *Annual Review of Neuroscience*, 26:381–410.
- Pouget, A., Deneve, S., and Duhamel, J. R. (2002). A computational perspective on the neural basis of multisensory spatial representations. *Nature Reviews Neuroscience*, 3:741–7.
- Pouget, A. and Sejnowski, T. J. (1994). A neural model of the cortical representation of egocentric distance. *Cerebral Cortex*, 4(3):314–29.
- Pouget, A. and Sejnowski, T. J. (1997). Spatial transformations in the parietal cortex using basis functions. *Journal of Cognitive Neuroscience*, 9(2):222–37.
- Pouget, A. and Snyder, L. (2000). Computational approaches to sensorimotor transformations. *Nature Neuroscience*, 3(supplement):1192–8.
- Pouget, A., Zemel, R. S., and Dayan, P. (2000). Information processing with population codes. *Nature Reviews Neuroscience*, 2:125–32.
- Pouget, A., Zhang, K., Deneve, S., and Latham, P. E. (1998). Statistically efficient estimation using population coding. *Neural Computation*, 10:373–401.
- Rao, R. P. N. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87.
- Rao, R. P. N., Olshausen, B. A., and Lewicki, M. S., editors (2002). *Probabilistic Models of the Brain: Perception and Neural Function*. MIT Press, Cambridge, MA.
- Reynolds, J. H. and Chelazzi, L. (2004). Attentional modulation of visual processing. *Annual Review of Neuroscience*, 27:611–47.
- Rosenblatt, F. (1962). *Principles of Neurodynamics*. Spartan Books, Washington, DC.
- Rothman, J., Cathala, L., Steuber, V., and Silver, R. A. (2009). Synaptic depression enables neuronal gain control. *Nature*, 457:1015–8.
- Rumelhart, D. E., McClelland, J. L., and The PDP Research Group, editors (1986). *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Volume 1: Foundations*. MIT Press, Cambridge, MA.
- Sahani, M. and Dayan, P. (2003). Doubly distributional population codes: Simultaneous representation of uncertainty and multiplicity. *Neural Computation*, 15(10):2255–79.
- Salinas, E. and Abbott, L. F. (1995). Transfer of coded information from sensory to motor networks. *The Journal of Neuroscience*, 15:6461–74.
- Salinas, E. and Abbott, L. F. (1996). A model of multiplicative neural responses in parietal cortex. *Proceedings of the National Academy of Sciences USA*, 93:11956–61.
- Salinas, E. and Sejnowski, T. J. (2001). Gain modulation in the central nervous system: where behavior, neurophysiology and computation meet. *The Neuroscientist*, 7(5):430–40.
- Salinas, E. and Thier, P. (2000). Gain modulation: a major computational principle of the central nervous system. *Neuron*, 27:15–21.
- Sanger, T. D. (1996). Probability density estimation for the interpretation of neural population codes. *Journal of Neurophysiology*, 76(4):2790–2793.
- Seilheimer, R. L., Rosenberg, A., and Angelaki, D. E. (2014). Models and processes of multisensory cue combination. *Current Opinion in Neurobiology*, 25:38–46.
- Shams, L. and Beierholm, U. R. (2010). Causal inference in perception. *Trends in Cognitive Sciences*, 14(9):425–32.
- Sherman, S. M. (2016). Thalamus plays a central role in ongoing cortical functioning. *Nature Neuroscience*, 19(4):533–41.
- Sherman, S. M. and Guillery, R. W. (1998). On the actions that one nerve cell can have on another: distinguishing “drivers” from “modulators”. *Proceedings of the National Academy of Sciences USA*, 95:7121–6.
- Shi, L. and Griffiths, T. L. (2009). Neural implementation of hierarchical bayesian inference by importance sam-

- pling. In *Advances in Neural Information Processing Systems*, volume 22, pages 1669–77. Curran Associates, Inc.
- Shipp, S. (2004). The brain circuitry of attention. *Trends in Cognitive Sciences*, 8(5):223–30.
- Smith, F. W. and Muckli, L. (2010). Nonstimulated early visual areas carry information about surrounding context. *Proceedings of the National Academy of Sciences USA*, 107(46):20099–103.
- Solbakken, L. L. and Junge, S. (2011). Online parts-based feature discovery using competitive activation neural networks. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1466–73.
- Spratling, M. W. (2008a). Predictive coding as a model of biased competition in visual selective attention. *Vision Research*, 48(12):1391–408.
- Spratling, M. W. (2008b). Reconciling predictive coding and biased competition models of cortical function. *Frontiers in Computational Neuroscience*, 2(4):1–8.
- Spratling, M. W. (2010). Predictive coding as a model of response properties in cortical area V1. *The Journal of Neuroscience*, 30(9):3531–43.
- Spratling, M. W. (2011). A single functional model accounts for the distinct properties of suppression in cortical area V1. *Vision Research*, 51(6):563–76.
- Spratling, M. W. (2012a). Predictive coding accounts for V1 response properties recorded using reverse correlation. *Biological Cybernetics*, 106(1):37–49.
- Spratling, M. W. (2012b). Predictive coding as a model of the V1 saliency map hypothesis. *Neural Networks*, 26:7–28.
- Spratling, M. W. (2012c). Unsupervised learning of generative and discriminative weights encoding elementary image components in a predictive coding model of cortical function. *Neural Computation*, 24(1):60–103.
- Spratling, M. W. (2013a). Distinguishing theory from implementation in predictive coding accounts of brain function [commentary]. *Behavioral and Brain Sciences*, 36(3):231–2.
- Spratling, M. W. (2013b). Image segmentation using a sparse coding model of cortical area V1. *IEEE Transactions on Image Processing*, 22(4):1631–43.
- Spratling, M. W. (2014). A single functional model of drivers and modulators in cortex. *Journal of Computational Neuroscience*, 36(1):97–118.
- Spratling, M. W. (2016). A review of predictive coding algorithms. *Brain and Cognition*, (in press).
- Spratling, M. W., De Meyer, K., and Kompass, R. (2009). Unsupervised learning of overlapping image components using divisive input modulation. *Computational Intelligence and Neuroscience*, 2009(381457):1–19.
- Spratling, M. W. and Johnson, M. H. (2003). Exploring the functional significance of dendritic inhibition in cortical pyramidal cells. *Neurocomputing*, 52-54:389–95.
- Spruston, N. (2008). Pyramidal neurons: dendritic structure and synaptic integration. *Nature Reviews Neuroscience*, 9:206–21.
- Spruston, N. and Kath, W. L. (2004). Dendritic arithmetic. *Nature Neuroscience*, 7(6):567–9.
- Stuart, G. and Häusser, M. (2001). Dendritic coincidence detection of EPSPs and action potentials. *Nature Neuroscience*, 4(1):63–71.
- Summerfield, C. and Egner, T. (2009). Expectation (and attention) in visual cognition. *Trends in Cognitive Sciences*, 13(9):403–9.
- Summerfield, C., Egner, T., Mangels, J., and Hirsch, J. (2006). Mistaking a house for a face: Neural correlates of misperception in healthy humans. *Cerebral Cortex*, 16(4):500–508.
- Vilares, I. and Kording, K. (2011). Bayesian models: the structure of the world, uncertainty, behavior, and the brain. *Annals of the New York Academy of Sciences*, 1224(1):22–39.
- Wacongne, C., Labyt, E., van Wassenhove, V., Bekinschtein, T., Naccache, L., and Dehaene, S. (2011). Evidence for a hierarchy of predictions and prediction errors in human cortex. *Proceedings of the National Academy of Sciences USA*, 108(51):20754–9.
- Yuille, A. and Kersten, D. (2006). Vision as Bayesian inference: analysis by synthesis? *Trends in Cognitive Sciences*, 10(7):301–8.
- Zemel, R. S., Dayan, P., and Pouget, A. (1997). Probabilistic interpretation of population codes. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems*, volume 9, pages 676–684. MIT Press.
- Zemel, R. S., Dayan, P., and Pouget, A. (1998). Probabilistic interpretation of population codes. *Neural Computation*, 10(2):403–30.